



ВОПРОСЫ НАУКИ И ОБРАЗОВАНИЯ

▶ **ELECTRONIC JOURNAL** • **ЯНВАРЬ 2024 № 1(173)** •

▶ **SCIENTIFIC-PRACTICAL JOURNAL**
НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

САЙТ ЖУРНАЛА: [HTTPS://SCIENTIFICPUBLICATION.RU](https://scientificpublication.ru)

ИЗДАТЕЛЬСТВО: [HTTPS://SCIENTIFICPUBLICATIONS.RU](https://scientificpublications.ru)

СВИДЕТЕЛЬСТВО РОСКОМНАДЗОРА ЭЛ № ФС 77-65699



ISSN 2542-081X



9 772542 081007

Вопросы науки и образования

№ 1 (173), 2024

Москва
2024





Вопросы науки и образования

№ 1 (173), 2024

НАУЧНО-ТЕОРЕТИЧЕСКИЙ ЖУРНАЛ
HTTPS://SCIENTIFICPUBLICATION.RU
EMAIL: TEL9203579334@YANDEX.RU

Издается с 2016 года.

Журнал зарегистрирован Федеральной службой по надзору в сфере связи,
информационных технологий и массовых коммуникаций (Роскомнадзор)
Свидетельство ПИ № ФС77 – 65699

Вы можете свободно делиться (обмениваться) — копировать и распространять материалы и создавать новое, опираясь на эти материалы, с **ОБЯЗАТЕЛЬНЫМ** указанием авторства. Подробнее о правилах цитирования:
<https://creativecommons.org/licenses/by-sa/4.0/deed.ru>

ISSN 2542-081X



© ЖУРНАЛ «ВОПРОСЫ НАУКИ И ОБРАЗОВАНИЯ»
© ИЗДАТЕЛЬСТВО «НАУЧНЫЕ ПУБЛИКАЦИИ»

Содержание

ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ	4
<i>Марк Б.И.</i> REPRESENTATIONAL STATE TRANSFER. ВВЕДЕНИЕ В ТЕХНОЛОГИЮ	4
ФИЛОЛОГИЧЕСКИЕ НАУКИ	22
<i>Равшанов М.</i> КОНЦЕПЦИИ ДЖАДИДИЗМА ЦЕНТРАЛЬНОЙ АЗИИ.....	22
<i>Равшанов М.</i> ОСНОВНЫЕ ИДЕЙНЫЕ ИСТОКИ ДЖАДИДИЗМА.....	25
<i>Равшанов М.</i> ИСТОРИЯ ДЖАДИДИЗМОВ В УЗБЕКИСТАНЕ.....	27
МЕДИЦИНСКИЕ НАУКИ	30
<i>Davlatov B.N., Maksudov B.M.</i> OPTIMIZATION OF THE TREATMENT TACTICS AND DIAGNOSTIC OF PATHOLOGICAL FRACTURES OF THE SPINE IN OSTEOPOROSIS.....	30
<i>Kodirov T., Odilov I.A., Ahmedov B.O.</i> CLINIC AND DIAGNOSTICS OF TRAUMATIC FRACTURES OF THE WRIST BONE AND OPTIMIZATION OF TREATMENT TACTICS	33

REPRESENTATIONAL STATE TRANSFER. ВВЕДЕНИЕ В ТЕХНОЛОГИЮ

Марк Б.И.

*Марк Богдан Игоревич – студент,
кафедра информатика и вычислительная техника,
Сибирский федеральный университет автоматизированные
системы обработки информации и управления,
г. Красноярск*

***Аннотация:** REST (Representational State Transfer) — это способ создания API с помощью протокола HTTP. На русском языке его называют «передачей состояния представления». Автор статьи описывает технологию на простых примерах для понимания начинающих обучение в сфере IT.*

***Ключевые слова:** REST, API, HTTP.*

Representational State Transfer или сокращенно REST. Что же это такое. В интернете полно статей, на разных языках о том, что это. Но многие трактовки этого понятия противоречат друг другу. Как же докопаться до сути.

Как по мне самый простой способ - разбить любое сложно понятие на составляющие, эти составляющие на их составляющие, пока не дойдем до основы.

Еще не плохой способ – коли информации имеется много, можно изучить ее большую часть, пускай даже если она противоречит другой. Когда мы изучим несколько точек зрения, можно их сравнить, пораскинуть немного своими мозгами, и выявить какую-нибудь одну свою, и дальше придерживаться её.

Для этого придется разобрать несколько примеров. Ах да, примеров... Без них примеров, конечно, никуда. Очень многие вещи нам зачастую помогают понимать именно примеры.

Пускай точка зрения, описанная в этой статье, будет дополнением к тем, которые Вы уже изучили, или Вам еще

предстоит изучить. Я буду использовать как можно меньше технических терминов, о них достаточно уже написано. Здесь я постараюсь изложить свое видение основных концепций, связанных с REST на простых примерах. Для того что бы в дальнейшем, при изучении Вами технической документации, процесс понимания сложных вещей был чуточку приятней. И если вдруг Вам останется, что-то не понятно, возможно все прояснится как раз таки в дальнейшем, при изучении REST уже на реальных примерах.

Кодекс разработчика

Самое распространенное и ключевое понятие REST — это **стиль**. Стиль взаимодействия, распределенного web приложения (или сложного, составного приложения).

Давайте немного поговорим об этом самом web приложении, и где оно там распределено. В обычной жизни приложением для нас является программка, скачанная на телефон или на планшет. Программа на рабочем компьютере. Или некоторые web сайты в интернете сейчас все чаще называют web версией приложения. Получается, что приложение это – приложение на телефоне, планшете или сайт в интернете? Да это является приложением, но зачастую – это всего лишь часть. Часть одного большого приложения, состоящего из нескольких таких частей.

Давайте разберем все на простом примере, чтобы было понятней. Предположим, Вы захотели узнать погоду. Вы скачали приложение на телефон, выбрали свой город и вот Вы знаете погоду на неделю вперед. Вроде бы простейшая обыденная ситуация, но мало кто задумывается, что для того, чтобы это стало возможно пришлось построить сложную систему, взаимодействующую между собой, то самое сложное приложение, распределенное по сети. Приложение, скачанное на телефон, или сайт, на котором Вы узнаете погоду это так называемая клиентская часть приложения, представляющую из себе некую красивую оболочку для отображения данных.

Сами данные берутся из скрытой от пользователя части приложения, которую называют серверной. Вся обработка и

хранение информации происходит там. Серверная часть приложения может так же делиться на части, состоять из нескольких приложений (серверов). Именно поэтому распределенное веб приложение и называют сложным. А распределенным оно считается, потому что территориально его компоненты могут находиться абсолютно в разных местах.

Со сложным распределенным приложение немного разобрались. Теперь давайте разбираться со стилем. Что это за стиль такой REST. Существует стиль в одежде, архитектурный стиль, музыкальный стиль и так далее. И что между всем этим общего? А то, что в любом стиле есть свои правила, как этому стилю соответствовать. И собственно совокупность этих правил и называют стилем. Т.е. получается, что и REST — это совокупность правил? В точку! REST это совокупность правил организации взаимодействия тех самых компонентов распределённого приложения, о которых мы говорили выше. Что-то вроде кодекса, написанного разработчиками для разработчиков.

Отлично, с терминами мы разобрались. Осталось разобраться, что это за правила такие.

Индейка в сливочном соусе.

Представьте себе, что вы приходите в ресторан, садитесь за столик и вот вы готовы сделать заказ. К Вам подходит официант, и Вы сообщаете ему что на горячее Вы будете индейку в сливочном соусе, на десерт пирожное с черносливом, а пить вы будете лимонный компот. Не плохой выбор! Официант внимательно выслушивает Вас и удаляется на кухню. Там он в свою очередь сообщает повару, что тот должен приготовить для Вас жаркое из индейки, пирожное с черносливом и компот.

Спустя какое-то время, официант возвращается к Вам и приносит Ваши блюда.

Все это здорово, но, казалось бы, причем здесь REST? Давайте рассмотрим все что произошло выше, как будто Вы воспользовались приложением (сложным приложением)

разработанному по всем канонам REST, далее мы будем называть его REST системой.

И так, Вы хотите что-то получить от официанта, назовем это что-то – ресурс. Для этого вы обращаетесь к официанту, как бы запрашивая у него этот ресурс. Другими словами, делаете запрос. И, стало быть, если вы запрашиваете у официанта этот ресурс, значит он у него есть. В системе REST, тот кто запрашивает ресурс называется клиентом, тот кто имеет этот ресурс называется сервером.

Получается, Вы являетесь клиентом, официант сервером. Вы передаете ему запрос, в котором указываете какую еду Вы бы хотели сегодня отведать (получить в ответ). Представим сам запрос вот в таком формате.

```
{
    "горячее": "индейка в сливочном соусе",
    "десерт": "пирожное с черносливом",
    "напиток": "лимонный компот"
}
```

Так гораздо лучше. Передавать запрос Вы будете по *url-адресу*:

`http://ресторан/официант/прием_заказов`

Отлично, ну и теперь сразу же, давайте коснемся методов передачи запросов. Методом является одно ключевое слово, в начале запроса, которое указывает какое действие с желаемым ресурсом вы бы хотели произвести. В данном случае, т.к. вы еще и сами передаете какую-то информацию, методом передачи запроса будет метод POST. Подробнее о запросах, а точнее об их методах, мы поговорим чуть позже. А пока имеем запрос такого вида:

Адрес: `http://ресторан/официант/заказ;`

Метод: POST;

Тело запроса:

```
{
    "горячее": "индейка в сливочном соусе",
    "десерт": "пирожное с черносливом",
    "напиток": "лимонный компот"
}
```


};

Официант обрабатывает данные, и передает их дальше – повару, так же в виде запроса. Здесь, уже официант берет на себя роль клиентом, а повар является сервером. Давайте представим это в виде наглядной схемы.



Рис. 1. Наглядная схема.

Как мы видим, на схеме есть клиенты и есть серверы. Тут сразу же стоит отметить, что речь идет о ролях компонентов системы, обращающихся друг к другу. И не в коем случае нельзя отождествлять понятия клиента и сервера с клиентской и серверной частью приложения. В данном случае клиентской частью будет посетитель ресторана (крайний левый клиент). Все что правее – серверная часть.

Вот так выглядит схема работы приложения, построенного по принципу клиент-серверной архитектуры. А мы с вами имеем, первое правило организации REST, это **клиент-сервер**. Есть клиент, есть сервер. Они взаимодействуют между собой путем отправки запроса и получения ответа.

Наверное, Вы заметили, что на схеме присутствует еще и база данных. В сложных приложениях так называют хранилище данных. Конечно, перед тем как начать готовить Вам еду, повар должен заглянуть в холодильник и убедиться в том, что у него есть все необходимые продукты. Тут холодильник будет являться базой данных, и только повар может взаимодействовать с ней. Тут мы с вами подошли ко второму правилу REST - **многослойная архитектура**.

Могло бы оказаться так, что в холодильнике не оказалось бы нужных продуктов и тогда бы повар на запрос официанта

вернул бы вместо готового блюда свои сожаления, которые официант в свою очередь передал бы Вам. Ему бы пришлось предложить Вам что-нибудь другое, поужинать Вас остаться, чтобы вы сделали новый заказ. Причем, заметьте, клиент может взаимодействовать только со своим сервером, а сервер со своим клиентом. Повар не выйдет к Вам в гостевой зал и не сообщит Вам плохую новость лично. И его абсолютно не волнует, как дальше пойдет взаимодействие у Вас с официантом. Продуктов нет и все тут, дальше не его вотчина. Вот же нахал... В этом и заключается особенность такая особенность REST как слоистая архитектура.

Но, все-таки предположим, что сегодня все сложилось удачно, все продукты оказались в холодильнике (базе данных), и официант вернул вам положительный ответ в виде горячего и компота, а не сожалений. Тут вроде бы все понятно, идем дальше.

Ну и запросы у Вас ...

Теперь давайте подробнее разберемся с нашими запросами, а точнее с методами их передачи. Что это за методы такие. Придется немного пофантазировать.

Оставшись довольны трапезой, Вы решили вернуться в это заведение в следующий выходной. Неделя, на удивление, пролетела очень быстро, и вот Вы снова сидите за столиком. А номер Вашего столика, предположим - 4. Почему бы и нет? К Вам подходит официант и Вы делаете заказ. Снова представим его в виде запроса:

Адрес: `http://ресторан/официант/заказ;`

Метод: `POST;`

Тело запроса:

```
{  
    "горячее": "индейка в сливочном соусе",  
    "десерт": "пирожное с черносливом",  
    "напиток": "лимонный компот"  
};
```

Теперь давайте более детально проанализируем, что же происходит дальше. Официант принимает Ваш запрос, и

формирует новый запрос, для отправки дальше – повару. Этот запрос уже будет выглядеть следующим образом:

Адрес: <http://ресторан/повар/заказы>;

Метод: POST;

Тело запроса:

```
{
    "заказ для": "столик №4",
    "заказ":
    {
        "горячее": "индейка в сливочном соусе",
        "десерт": "пирожное с черносливом",
        "напиток": "лимонный компот"
    }
};
```

Особо внимательный заметит, что в конце адреса, по которому передает свой запрос официант указано - заказы, во множественном числе. Это и правильно, ведь посетителей ресторана много, как и заказов. Но как официанту с поваром ориентироваться во всех этих заказах? Для этого в запросе официанта добавился еще и номер Вашего столика. Это поле будет неким идентификатором или ID заказа. После чего, будет создан некий ресурс (заказ), доступный по адресу:

http://ресторан/повар/заказы/заказ_для_столика_№4

И все дальнейшие манипуляции с этим заказом, будут происходить именно по этому адресу.

Пока официант был на кухне передавая данные повару, Вы вспоминаете, что один Ваш приятель рассказывал Вам про чудесный суп с грибами, который он ел в этом ресторане. Как только в следующий раз официант попадает к Вам в поле зрения, Вы подзываете его к себе и сообщаете, что очень хотели бы попробовать еще и суп. Тут переданный Вами запрос будет иметь следующий вид:

Адрес: <http://ресторан/официант/заказ>;

Метод: PATCH;

Тело запроса:

```
{
    "первое блюдо": "грибной суп",
```

};

Метод этого запроса будет – РАТСН. Тем самым будет понятно, что к уже имеющемуся заказу, нужно добавить первое блюдо. Теперь официант как-то должен донести до повара информацию о том, что Вам нужно приготовить еще и суп. Как это сделать? Конечно же отправить запрос. Но этот запрос будет уже немного другим.

Адрес:

http://ресторан/повар/заказы/заказ_для_столика_№4;

Метод: РАТСН;

Тело запроса:

{

 "первое блюдо": "грибной суп",

};

Адрес отправки запроса уже будет следующий:

http://ресторан/повар/заказы/заказ_для_столика_№4

Вот для чего нам нужен был ID заказа. Когда официант будет передавать данные повару, ему будет понятно, что речь идет об уже имеющемся заказе для столика №4.

Отлично, теперь повар поймет, что ему нужно приготовить суп и отдать его официанту, обслуживающему столик №4.

Ну что ж, повар приступил к приготовлению, но Вы немного поразмыслили поняли, что грибной суп — это не совсем то, чего бы Вам сегодня хотелось. Что с гораздо большим аппетитом Вы бы съели суп с говядиной. И, пожалуй, при таком раскладе пирожное с черносливом уже не поместится в Вашем желудке. Придется опять изменять заказ. Не беда, в этом заведении Вы можете себе такое позволить. Легкий жест рукой и официант снова у Вашего столика, готовый выслушать Ваши новые пожелания. В этой ситуации придется уже сделать два запроса.

Первый, методом РАТСН для изменения первого блюда:

Адрес: <http://ресторан/официант/заказ>;

Метод: РАТСН;

Тело запроса:

{

 "первое блюдо": "суп с говядиной"

```
};
```

Второй, еще не знакомым нам методом DELETE для отмены десерта:

Адрес: <http://ресторан/официант/заказ;>

Метод: DELETE;

Тело запроса:

```
{  
    "десерт": "пирожное с черносливом"  
};
```

Отлично, Вы отказались от десерта, и перезаказали суп. Теперь Вы точно останетесь довольны ужином.

Официант же, спешит на кухню сообщить повару, чтобы тот не торопился класть в суп грибы. Там он также может передать два запроса повару:

Первый:

Адрес:

http://ресторан/повар/заказы/заказ_для_столика_№4;

Метод: PATCH;

Тело запроса:

```
{  
    "первое блюдо": "суп с говядиной"  
};
```

Второй:

Адрес:

http://ресторан/повар/заказы/заказ_для_столика_№4;

Метод: DELETE;

Тело запроса:

```
{  
    "десерт": "пирожное с черносливом"  
};
```

Но есть и альтернативный вариант. Тут то и появляется на сцене еще один метод – PUT. Метод PUT может целиком заменить уже имеющийся ресурс, в нашем случае речь идет о заказе.

И так, официант обращается по адресу:

http://ресторан/повар/заказы/заказ_для_столика_№4;

И методом PUT передает запрос:

```
{  
    "первое блюдо": "суп с говядиной",  
    "горячее": "индейка в сливочном соусе",  
    "напиток": "лимонный компот"  
};
```

Отлично. Старый заказ будет удален, а вместо него появится новый. Теперь повар поймет, что для столика №4 ему не нужно готовить пирожное. А вместо шампиньонов в суп он положит свежую говядину.

Существует так же метод GET, пример с ним мы рассмотрим чуть позже. На самом деле методов запросов существует больше. Мы рассмотрели только самые основные.

Что ж, и в этот раз все блюда Вам благополучно были доставлены. Вы остались довольны, и конечно же решили вернуться сюда, в следующий свой выходной.

На самом деле в этой главе мы рассмотрели не только методы передачи запроса, но еще и коснулись одного из самых главных правил REST это – единого интерфейса. Это правило достойно отдельной главы, поэтому о нем мы поговорим дальше.

Один как все и все как один

И так, единый интерфейс. Что это? В данном случае, слово интерфейс можно отождествить как способ взаимодействия. Т.е. дальше речь пойдет о едином способе взаимодействия. Конечно же, в контексте REST взаимодействия компонентов приложения, которыми как мы уже говорили являются клиент и сервер. Хм, что мы имеем теперь? Одно из правил REST- единый интерфейс – это правило единого способа взаимодействия компонентов приложения. А само понятие REST это совокупность правил организации взаимодействия тех самых компонентов распределённого приложения. Т.е. получается, что одно из набора правил взаимодействия – это правило взаимодействия. Масло масляное. Умозаключение конечно логичное, но смысловой нагрузки пока в себе не несет никакой. Давайте разбираться.

Вернемся к нашему вежливому официанту. Как мы уже сказали, совокупности правил взаимодействия REST, есть отдельное правило общения, которое так и называется правило взаимодействия. Официант компонент системы? Компонент. Значит и для него применимо это правило. Мы уже коснулись ролей взаимодействия, зон ответственности, но еще не говорили о поведении. Да, единый интерфейс – это что-то вроде правила поведения во время взаимодействия. Дак как же должен вести себя наш официант? Он должен быть учтив, внимателен, вежлив. Он должен знать кому и что ему можно и нужно говорить. Когда он подойдет к Вашему столику, он предложит Вам сделать заказ и не начнет рассказывать Вам о том, как правильно играть в гольф, или как готовить круассаны с клубникой. Ему совершенно не важно, кто будет его клиентом. С любым своим клиентом он будет взаимодействовать одинаково учтиво – выслушивать заказ, вежливо отвечать на вопросы, приносить еду и напитки. Так же повару – неважно для какого столика он будет готовить еду, какой официант принес ему заказ. Повар будет стараться приготовить одинаково для всех.

В этом и есть прелесть REST. Компоненты REST системы стандартизированы, их можно легко заменить другими такими же компонентами, главное, чтобы они знали, как им взаимодействовать друг с другом. Если вдруг посещаемость ресторана будет расти, можно увеличить количество официантов. Ведь правила поведения официантов везде одинаковые. Нужно только показать новенькому официанту где кухня, и он готов к работе. При этом другие элементы системы, затронуты не будут, и никакой работы над ними проделывать не нужно.

Представим еще, что наш любимый ресторан стал предлагать такую услугу как – еда на вынос. Тут потребуются новые компоненты – доставщики еды. Они будут общаться с клиентской частью по средством телефонной связи или мессенджеров, но передавать заказ повару они будут абсолютно так же, как и официанты. Отправлять запрос:

Адрес: `http://ресторан/повар/заказы`;

Метод: POST;

Тело запроса:

```
{
  "заказ для": "доставщик Иван Иванов",
  "заказ":
  {
    "горячее": "индейка в сливочном соусе",
    "десерт": "пирожное с черносливом",
    "напиток": "лимонный компот"
  }
};
```

Тем самым создавая ресурс:

`http://ресторан/повар/заказы/заказ_на_вынос_Иван_Иванов`

;

Процесс изменения размеров системы называется — *масштабированием*. Масштабирование — это один из неотъемлемых плюсов REST систем. В любой момент можно легко изменять размеры серверной части, в зависимости от потребностей клиентской. И во многом это благодаря правилу единого поведения. Замечательное правило! О нем еще так же стоит сказать, что само оно включает четыре подправила, для того чтобы было проще следовать ему.

Первое – ресурсоемкость или ресурсный подход.

Суть этого правила заключается в том, хоть у каждого ресурса и есть свой идентификатор (`заказ_для_столика_№4`) он должен быть включен в обобщенное понятие (`заказы`), для того чтобы мы знали к какой группе ресурсов относится это ресурс.

Второе – манипуляция ресурсами посредством представлений.

Я бы даже сказал посредством перепредставлений, или если быть еще точнее перепредоставлений. В оригинале это правило называется Manipulation of Resources Through Representations. Сама аббревиатура REST - Representational State Transfer. И самое важное тут понять, что же это за Representations и Representational.

Representations – переводится как представление. У клиента есть свое представление о ресурсе. Оно может меняться. Как мы помним из нашего примера, мы могли несколько раз изменять свой заказ. Для этого мы озвучивали свое представление того, как должен выглядеть наш заказ официанту, а тот в свою очередь повару.

Тогда что же такое Representational? В контексте с REST, пожалуй, можно дать значение этому слову – актуальный или наиболее точный.

Из чего следует, что REST можно перевести как – передача актуального состояния.

Но тут стоит отметить – изменять ресурс может только тот у которого есть на это право. Официант, обслуживающий столики № 3, 4 и 7 не может прийти к повару и изменить заказ для столика № 1.

Третье – информативные сообщения. (Абсолютно точные сообщения.)

Когда клиент обращается к серверу он должен быть абсолютно точным в своих обращениях.

Если за столиком №3 кто-то закажет моккачино, официант не должен придя к повару крикнуть ему: «Мне моккачино, и побыстрее!». Повар может подумать, что официант устал и хочет выпить кофе дабы восполнить силы. На скорую руку он сварит не самый лучший в его практике напиток, положит туда побольше кофе, дабы заряда бодрости официанту хватило на подольше. Да и еще нальет все это в железную большую кружку, т.к. чистой чашки под рукой не окажется.

Вместо этого официант должен прийти к повару и сказать: «Заказ для третьего столика. Моккачино.» Вот тогда повар правильно поймет официанта. Приготовит свой лучший напиток и нальет его в чистую фарфоровую чашку.

Четвертое - гипермедиа как движущая сила состояния приложения. Hypermedia as the Engine of Application State или сокращенно (HATEOAS). Об этом правиле давайте поговорим чуть позже.

Официант не был бы официантом будь у него состояние...

В каждый Ваш приход в ресторан, Ваше общение с официантом начинается с чистого листа. У Вас не получится сказать ему: «Я буду тоже, что и в прошлый раз». Официант не запоминает кто и что заказывал в прошлый свой приход. Поэтому если Вы захотите повторить свой заказ, Вам придется потратить немного времени и снова озвучить весь список желаемого. Это и будет называться *Stateless или отсутствием состояния*, что в свою очередь является еще одним правилом REST. Может это конечно не очень удобно для клиента, помнить, что он заказывал в прошлый раз приходится именно ему. Да и времени, чтобы повторять свой заказ снова, уйдет гораздо больше, нежели просто скажи Вы: «Мне как обычно!». Но, давайте вспомним о том, что официант обычно обслуживающий Ваш столик, хоть и добросовестный работник, но все же человек, и может попросту заболеть. И его заменит другой. И вот он, этот другой официант, уже точно не будет помнить, что Вы там ели в прошлый раз. Ему не придется звонить официанту, который обслуживал Вас в прошлый раз и узнавать Ваш прошлый заказ. Пока тот вспомнит... Да и телефона может не оказаться под рукой. Гораздо проще и быстрее будет если официант просто запишет Ваш заказ и отнесет повару, согласитесь? А если в следующий Ваш приход и это официант не выйдет на работу по каким-то причинам, его с легкостью заменит любой другой, который будет не занят. Это еще один критерий, благодаря которому возможно масштабирование масштабированием.

Поскольку заказ Вам придется делать по новой, Вы решили изучить меню. Ваше внимание привлек салат с сельдереем. Подозвав официанта, Вы решили поинтересоваться у него, с чем же этот салат.

Вы снова делаете запрос, но в этот раз уже долгожданным методом GET. Ведь в этот раз, Вы сами уже не передаете никаких данных. Выглядеть это будет примерно так:

Адрес: [https://ресторан/официант/составы блюд/салат с сельдереем](https://ресторан/официант/составы_блюд/салат_с_сельдереем);

Метод: GET

Как видите, тут уже нет никакого тела запроса. Мы хотим получить информацию ничего не передав. А то, относительно чего мы хотим получить информацию, мы указываем в адресной строке.

Теперь официанту нужно дать вам ответ, в виде рецепта салата. Официант, к сожалению, не знает рецепта. Он удаляется на кухню, чтобы поинтересоваться у повара – из чего же тот готовит этот салат. И спустя какое-то время официант возвращает Вам ответ:

```
{  
    "салат с сельдереем": "курица, сельдерей,  
перепелиное яйцо, оливковое масло"  
}
```

Такой себе салат, не правда ли? Но о вкусах не спорят, хоть Вы этот салат решили не заказывать, а вот за соседнем столиком, кто-то услышала про этот салат, и тоже решил поинтересоваться у официанта, из чего-же готовят этот салат. Официант опять удаляется на кухню, он уже забыл рецепт. Ах этот неряшливый официант... Но повар твердо знает, что он кладет в свой салат, и спустя время официант возвращается к клиенту с рецептом салата. Салат с сельдереем становится все популярнее. Теперь уже в другом конце зала, кто-то решил попробовать салат с сельдереем. И вот официант снова направляется на кухню, чтобы узнать рецепт. Согласитесь, не самая лучшая ситуация? Какой же может быть выход из нее. Пожалуй, рецепт салата, это неизменяемые данные, или не так часто изменяемые. Поэтому официанту вполне было бы не плохо запомнить его. Тогда он сможет озвучивать клиентам рецепт салата мгновенно, не тратя свое время на походы на кухню, да и повара не будет отвлекать по чем зря. Запоминание рецепта салата в этой ситуации будет называться кэшированием. Кэширование — это еще одно правило REST.

Но тут стоит отметить одну особенность. Кэшировать стоит только те данные, которые точно не изменятся, а если вдруг они поменяются, то нужно так же перезаписать их в кэше.

Вдруг у повара не окажется курицы, и он решит добавить в салат вместо нее рыбу. Согласитесь, будет не очень хорошо, если гостю принесут салат с рыбой, в то время как официант пообещает ему салат с курицей.

Теперь давайте вернемся к пропущенному понятию **НТЕОАС**. Придя в какое-нибудь заведение, и поинтересуйся там рецептом салата, официант в ответ сообщил бы вам только рецепт, и ничего более. Но в этой статье мы рассматриваем заведение, где взаимодействие компонентов происходит согласно всем правилам REST. И что это будет значить Вас?

Когда Вы поинтересуетесь у официанта из чего же будет приготовлен салат, в ответ Вы услышите не только сам рецепт салата, но и то, какое еще блюдо Вы можете заказать в дополнение к этому салату, с каким напитком его лучше попробовать. А также другие блюда, которые Вы бы могли попробовать, не окажись этого салата в наличии. Или информацию о тех блюдах, о которых по мнению официанта, Вам очень важно было бы знать.

Согласно правилу **НТЕОАС**, сервер в ответ на запрос возвращает не только запрошенную информацию или ресурс, но и отсылки к другим ресурсам, чтобы клиент мог в них ориентироваться.

Не плохо правда? Теперь Вы можете заказать, что-то еще. Или вообще изменить свой заказ. Вроде бы это хорошее правило, но не всегда оно работает во благо. Сильно расторопный клиент может глубоко задуматься, что же теперь ему стоит заказать. Выбор окажется настолько мучительным, что чтобы его сделать потребуется много времени. Поэтому тут, официанту стоит все же хорошо подумать прежде, чем формировать свой ответ.

Официант, дичь!

И снова Вы в своем любимом ресторане. В этот раз Вы заказали не жаркое из индейки, а блюдо из дикой куропатки. Особенность этого блюда в том, что его нужно есть с пылу с жару. Как же этого добиться? Ведь пока официант дойдет до повара, пока принесет блюдо Вам. А если в этот момент

будет занят обслуживанием другого столика... Пройдет много времени, блюдо остынет. К счастью для Вас гении кулинарного дела придумали выход – заливать блюдо алкоголем и поджигать. Отличное решение, но согласитесь следовать официанту через весь зал с горячей куропаткой не самое лучшее решение. Хм... А ведь можно просто снабдить официанта всем необходимым, скажем – блюдом залитым алкоголем и спичками, и поджиг выполнить уже непосредственно в момент подачи, скажем так – на стороне клиента.

Подобным образом и работает последнее правило REST – код по запросу. Допустим нам нужно выполнить какой-то сценарий на стороне клиента. Не обязательно выполнять его еще и на сервере, не правда ли? Можно отправить код на клиентскую часть приложения, и там уже выполнить его. Примерно так же, как поджечь куропатку под носом у клиента.

Подведем итоги

Что ж, мы выяснили, что для того, чтобы организовать REST сервис, необходимо придерживаться следующих шести правил:

1. Клиент-серверная архитектура. Вы клиент, официант - сервер. Вы можете обмениваться друг с другом информацией или чем-то еще. Вы – клиентская часть. Официант, повар и холодильник – серверная часть.

2. Слоистая архитектура. Клиентов и серверов может быть много. Промежуточные элементы являются как клиентом для своего сервера, так и сервером для своего клиента. Каждый клиент может общаться только со своим сервером, а сервер со своим клиентом. Клиент не может общаться с сервером своего сервера. Помните, повар не принесет Вам жаркое сам лично.

3. Единый интерфейс. Способ общения компонентов обозначен определенным стандартом, манерами поведения.

4. Никто не хранит состояние. Новый сеанс общения – чистый лист.

5. Кэширование. Все-таки какую-то часто используемую информацию хранить можно.

6. Код по запросу. Если нужно выполнить сценарий на стороне клиента, можно предоставить ему для этого все необходимое, пускай выполняет...

И так, система REST это распределенное в сети приложение, спроектированное согласно кодексу REST. Хотя кодекс это и не свод жестких законов, а правил, нарушать эти правила нельзя. Разве что есть одно не обязательное к исполнению правило – код по запросу.

Теперь обладая общими представлениями о REST, Вы можете приступить к изучению технической документации по этой теме, чтобы понимать работу системы в реальных ситуациях.

Список литературы

1. *Акинин Ю.Р., Барабанов А.В., Гребенникова Н.И.* Быстрое создание Rest API сервиса на основе облачных технологий Azure // Вестник ВГТУ. 2012. №12-1. URL: <https://cyberleninka.ru/article/n/bystroe-sozdanie-rest-api-servisa-na-osnove-oblachnyh-tehnologiy-azure> (дата обращения: 28.01.2024).
2. *Безрук П.А.* Разработка системы распределенного мониторинга компьютерной сети на основе rest API // Актуальные проблемы авиации и космонавтики. 2017. №13. URL: <https://cyberleninka.ru/article/n/razrabotka-sistemy-raspredeleennogo-monitoringa-kompyuternoy-seti-na-osnove-rest-api> (дата обращения: 28.01.2024).

КОНЦЕПЦИИ ДЖАДИДИЗМА ЦЕНТРАЛЬНОЙ АЗИИ Равшанов М.

*Равшанов Махмуд – профессор, доктор филологических наук,
Навоийский государственный горный университет,
г. Навои, Республика Узбекистан*

Аннотация: эффективные формы интегрирования реформаторов Центральной Азии была национальная периодическая печать джадидизмов. Периодическая печать формировались в конце XIX в. Этот публицистический процесс получил свое бурное развитие в начале XX в. Концепции джадидов пропагандировались во всей Центральной Азии.

Ключевые слова: концепция, джадидизм, Центральная Азия, печать джадидов.

УДК.812.111

Органы печати джадидов являлись, по мнению просветителей, важнейшим инструментом идейного воздействия на сознание широких слоев общественности данного времени. туркестанские прогрессисты “без национальной печати нет самой нации”. Печать джадидов для них была своеобразной трибуной пропаганды. Через этого они доводили до масс свои идеи **обновления общества** [1, с. 12]. В течение первых десятилетий X века в Центральной Азии, при их непосредственном участии, были, открыты газеты и журналы джадидов. На основе новой волны просветительства. Эти газеты и журналы уже вели пропаганду широкой программы политических реформ нового общества в Центральной Азии. В сложности данного времени их было более 15 названий [2, с. 37-41]. Этот процесс общественности была уже большая идеологическая сила. Получил в своем развитии огромный диапазон. От просветительских идей до политических опубликованы многих статей. Национальная печать данного времени

джадидов наглядно продемонстрировала наличие в крае нового типа человека, с прогрессивным мышлением, заботящегося о благе нации и борющегося с косностью, отсталостью, невежеством.

По данным источников история национальной прессы началась 27 июня 1906 г., когда ташкентские прогрессисты во главе с Исмаилом Обидовым (Габитовым) начали издание первой газеты на узбекском языке “Тариккий” (“Прогресс”). Руководитель этих общественных процессов являлся Исмаил Габитов. Он родился 19 июля 1881 г. в городе Кульджа в Восточном Туркестане в семье торговца. Его отец Шарафуддин Габитов, был образованным человеком своего времени. Он знал многие восточные языки, долгое время служил в посольстве России в Кульдже. После его смерти в 1890 г. (по некоторым данным в 1887) [3, с. 43-67] Исмаил Габитов при поддержке брата матери Рашида Сулейманова переезжает с матерью в Казань. Здесь до 1900 года будущий просветитель джадидов живет и учится в реальном училище. После окончания данного училища Исмаил Габитов переезжает в Москву и поступает в школу зодчества. После революции 1905 г. он уезжает в Туркестанский край. Уже в Ташкенте под сильным влиянием от происходивших вокруг революционных событий, вместе с прогрессивными представителями просвещения и культуры данного времени Мунаввар кары Абдурашидовым, Абдуллой Авлони, Махмудходжой Бехбуди и другими начинает выпускать первый номер которой вышел 14 (27) июня 1906 г.

Газета сразу же начинает пропагандировать идеи свободы, равенства, дружбы народов. В одной из своих редакционных статей, И. Габитов писал о том, объявлено о свободе слова, но реальная “свобода” в Туркестане по-прежнему существует лишь на бумаге. Значительное внимание на страницах газеты уделялось пропаганде межэтнической и межконфессиональной толерантности. Так, например, в статье “Ташкент, 12 августа”, была подвергнута резкой критике по разжиганию межнациональной вражды. “Они унижают и бранят нашу религию и святимый Коран. Пишут

они об этом на страницах своих [4, с. 61-85] газет и распространяют их по всему миру. Стравливают людей на основе межнациональных отношений, противопоставляя людей друг другу. Народ Туркестана по-прежнему заставляют унижаться и преклоняться перед чиновниками, подвергая телесным и моральным наказаниям”. Конечно же газета уделяла большое внимание проблемам просвещения. В первом и во втором номерах газеты была опубликована статья Мунаввар кары Абдурашидханова “Наши невежества являются сложными невежествами”, в которой говоря о необходимости реформирования школ и медресе, автор писал: “там ... должны преподавать Тараққий. 1906,12 август. 90 Тараққий. светские науки, географию, историю, математику и другие предметы не зная эти предметы, наши дети до сих пор остаются без нужных знаний и образований [5, с. 60].

Таким образом, концепции джадидов были в образовательном процессе внедряют инновационные и новые методы преподавания.

Список литературы

1. *Турдиев Ш.* Туркестанские прессы джадидов. Т.6 2003. С.12
2. Редактора газеты называли - “Тараққий” (Прогресс). О судьбе восточного просветителя Исмаила Габитова// 1917 г. были организованы такие печатные органы, как “Хуррият”, “Нажот”, “Кенгаш”, “Турон”, “Улуғ Туркистон”.
3. *Абдуазизова Н.А.* Туркестан матбуоти тарихи. - Тошкент: Академия, 2000. - 43-76
4. *Ўзганбоев Ҳ.* Ўзбек маърифатпарварлик адабиётининг тараққиётида вақтли матбуотнинг ўрни (“Туркистон вилоятининг газети”, “Тариққий”, “Самарқанд”, “Туркистон” рўзномалари асосида). - Тошкент: Фан, 1993. 85-б.

5. *Тараққий*. 1906, 14 июнь. 7 0 газету на узбекском язмке “Тараккий” (“Прогресс”), 60-б.

ОСНОВНЫЕ ИДЕЙНЫЕ ИСТОКИ ДЖАДИДИЗМА Равшанов М.

*Равшанов Махмуд – профессор, доктор филологических наук,
Навоийский государственный горный университет,
г. Навои, Республика Узбекистан*

Аннотация: *джадиды стремились овладеть достижениями восточной и западной культур, учились и призывали других к тому же. Побывав за рубежом, просветители сумели разглядеть и сопоставить уровни развития культуры и образования различных стран.*

Ключевые слова: *джадиды, «Таржимон», школы «усули джадид».*

УДК.812.111

Основные идейные истоки джадидизма рассматриваются идейные предпосылки джадидизма в Центральной Азии. В данной статье утверждается, что огромное влияние на формирование воззрений бухарских джадидов оказал крупный идеолог и родоначальник просветительского движения в Центральной Азии Ахмад Дониш (1827-1897), на произведениях и идеях которого воспитывалось молодое поколение в Бухаре на рубеже XIX и XX веков.

С. Айни отмечает что, именно его труды производили среди передовой части студентов медресе «настоящую духовную революцию... Воздействие на нас его книги было столь сильным, что наши взгляды на тогдашнюю жизнь изменились почти совершенно». В данной статье отмечается, что большое влияние на развитие общественного сознания, просвещения и педагогической мысли в Туркестанском крае оказали виднейшие узбекские просветители-демократы в конце XIX – начале XX столетия Муками (1850-1903),

Фуркат (1858-1908), Завки (1853-1921), каракалпакский поэт Бердах (1827-1900) и др.

Другой школой вольнодумия для среднеазиатской молодёжи стала татарская общественно-политическая мысль и в первую очередь просветительство поволжских татар XIX века. Как указывает татарский исследователь С.Г. Батыев, «джадидское движение заимствовало идеи перестройки мусульманского образования в значительной мере у реформаторов Курсави, Хусаина Фаизханова и Марджани».

Подчеркивается, что весомый вклад в развитие джадидских идей среди местного населения Туркестана, Бухары и Хивы внес основоположник татарского джадидизма Исмаил Гаспринский (1851-1914). Отмечается идеи джадидизма, что влияние прогрессивных идей И. Гаспринского наблюдается в творчестве джадидских литераторов М. Бехбуди, Васли, Аджзи, Мирзо Сироджа, Абдуррауфа Фитрата, Ходжи Муина, Вадуда Махмуди, Хатоии Самарканди, Фахриддина Роджи, Абдукадыра Шакури и др.

В развитии джадидизма в Центральной Азии особую роль сыграли различные журналы и газеты, издаваемые в России и зарубежных странах Востока, прежде всего «Таржимон» («Переводчию»), «Вақт» («Время»), «Шуро» («Совет»), «Мулло Насреддин» и др., о чем свидетельствуют материалы [1, с. 156-163].

Ранний историк Е. Фёдоров писал, что «течение под названием «джадидчи» зародилось в Центральной Азии под непосредственным влиянием газеты «Таржимон». Как видно из архивного документа, в 1914 году газету «Таржимон» выписывали жители города Худжанда (Ходжент) Алимбаев, Мухаммад Юсуф Мулла Хальмухамедов и Мир Исмаил Мир Захидов, каждый по 2 экземпляра. Однако тюрко-татарские прогрессивные издания оказали влияние и на распространение идей пантюркизма в Центральной Азии [2, с. 62-63].

Таким образом, на основе изложенного в данном разделе диссертант приходит к выводу, что в формировании

джадидских идей в Центральной Азии большое влияние оказали крупнейший представитель среднеазиатского просветительства — Ахмади Дониш, узбекские просветители Муками, Фуркат, Завки, религиозные реформаторы Поволжья Абу Наср Курсави, Шахобиддин Марджани, Хусаин Фаизханов, лидеры реформаторского движения стран Востока Зайн ал-Абидин Мароги, Сайд Джамолиддин Афгани, Мухаммад Абдо, основатель татарского джадидизма Исмаил Гаспринский и др. [3, с. 55-67].

Таким образом, немало важное влияние на становление мировоззрения джадидов оказала прогрессивная мусульманская печать России и стран Востока рубежа XIX-XX вв.

Список литературы

1. *Вексельман М.И.* Складывание национальной буржуазии в Центральной Азии в конце XIX - начале XX века — История СССР - 1957.-№3.-С.156-164.
2. *Айни С.* Коротко о моей жизни,- Сталинабад, 1958 - С. 62-63.
3. *Батыев С.Г.* Татарский джадидизм и его эволюция // История СССР - 1964,- № 4 - С. 55.

ИСТОРИЯ ДЖАДИДИЗМОВ В УЗБЕКИСТАНЕ

Равшанов М.

*Равшанов Махмуд – профессор, доктор филологических наук,
Навоийский государственный горный университет,
г. Навои, Республика Узбекистан*

Аннотация: возникновение джадидизма несет в себе идею единства тюркского мира. Основатель этого движения жадидизма Исмаил Гаспралы родился в Крыму. На своем опыте он осознал, как важно сохранить тюркскую идентичность в условиях колониализма. Посредством тюркоязычного печатного издания «Таржимон» ему

удалось распространить свои идеи по всему Туркестану и тюркско-язычной части Российской империи.

Ключевые слова: *джадидизм, прогрессивные идеи.*

УДК.812.111

Как свидетельствует «Джадиды возникли как группа, которая стремилась найти ответы на вопросы своего времени. Это такие вопросы, как почему мы (как регион) были завоеваны и как выбраться из этой ситуации. Панацеей были образование и национальное единство», — объяснил в интервью профессор истории Карлтонского колледжа (США) Адиб Халид. Идеи Исмаила Гаспралы вдохновили муфтия Самарканда, просветителя Махмудходжа Бехбуди. Имея право на фетву, он доказал, что образование и искусство не только не противоречат канонам ислама, но и наоборот — очень приветствуются [1, с. 34-46].

Таким образом джадидизм (обновление) в Туркестане возникло как движение просвещенных мусульман. В первые 30 лет развития джадидизма удалось достигнуть больших успехов: открытие школ, применяющих новые методики, создание национальных театров, распространение прогрессивных идей, формирование ответственной бизнес-элиты. Они встречали сильное сопротивление как со стороны колонизаторов, так и со стороны феодальной отсталой части населения. Эмир Бухарский Саид Алим Хан был настроен враждебно по отношению к активистам этого движения. В результате заговора Махмудходжа Бехбуди был похищен и казнен в 1919 году. С этого момента начинается новый этап в истории этого движения джадидизма [2, с. 54-67].

Часть джадидов покинула свой край, другая часть примкнула к басмаческому движению. Однако некоторые надеялись при советской власти продолжить свою просветительскую деятельность. Ведь провозглашенные идеи, такие как «Землю крестьянам», «Заводы рабочим» и «Право нации на самоопределение», а главное «Тотальная борьба с безграмотностью», открывали для джадидов еще большие

возможности. В 1920 году образовалась Бухарская народная советская республика, которую возглавил 24-летний джадид — Файзулла Ходжаев. Республика просуществовала 4 года, и ее можно назвать государством джадидов. Она имела свой флаг, свою конституцию и даже свою армию.

За 4 года Файзулле Ходжаеву удалось многое. Например, около 1000 молодых людей были направлены на учебу в Россию, Германию, Турцию и Азербайджан. Позже они составят костяк интеллигенции бывшего Туркестана, на месте которого образовалось 5 республик. Однако все те, кто учился в Германии и Турции, впоследствии были практически истреблены. Одних обвинили в пантюркизме, других — в шпионаже в пользу Запада. Но, несмотря на это, с 1920 года по 1937-й джадидам удалось сделать многое — осуществить свою мечту, а главное взрастить приемников [3, с. 125].

Трагическая гибель джадидов в СССР была неминуема. Но, к счастью, несмотря на усердные попытки советского государства нивелировать память о них, это оказалось невозможным. На кладбищах в Турции можно встретить много надгробных памятников с надписью «Здесь покоится туркестанец». Идеи джадидов выдержали испытание временем и оказались востребованными для всего тюркского мира [3, с. 12].

Таким образом, задачи подчеркнуть их роль в истории Узбекистана и тюркской культуры ставили перед собой организаторы международной научной конференции «Джадида: идеи национальной идентичности, независимости и государственности».

Список литературы

1. *Имамназаров М.* «Миллий маънавиятимизнинг такомил боскичлари». Т., «Иктисод молия», 2008. Основ.
2. *Қосимов Б.* «Миллий Уйғониш». Ташкент. 1997. 125 с.
3. *Турдиев Ш.* Туркистанские прессы джадидов. Т.6 2003. С. 12.

OPTIMIZATION OF THE TREATMENT TACTICS AND DIAGNOSTIC OF PATHOLOGICAL FRACTURES OF THE SPINE IN OSTEOPOROSIS

Davlatov B.N.¹, Maksudov B.M.²

¹*Davlatov Baxodir Nabijanovich - Doctor of medical sciences,
Associate Professor;*

²*Maqsudov Baxtiyor - Independent researcher,
Andijan State Medical Institute,
Andijan, Republic of Azerbaijan*

Abstract: *optimization of the treatment tactics and diagnostic of pathological fractures of the spine in osteoporosis. This article presents the experiences of optimization of the treatment tactics and diagnostic of pathological fractures of the spine in osteoporosis.*

Keywords: *optimization, fracture, osteoporosis, calcium, MRT, treatment tactics, spine, pathological fracture diagnosis, surgery.*

Rentgenografiya osteoporozni erta bosqichda aniqlay olmaydi. Osteoporoz komyuter tomografiyasi (KT) yoki magnit-rezonans tomografiya (MRT) yordamida suyak to‘qimasida suyak zichligi pasaygan o‘choqlar shaklida aniqlanadi. Tadqiqot ma’lumotlarini vaqt bo‘yicha taqqoslash katta ahamiyatga ega. Zamonaviy kompyuter tomografiyalari suyakning uch o‘lchamli modelini olish, suyak to‘qimalarining zichligini o‘lchash, uning hajmini va ular orasidagi bo‘shliqni o‘lchash, suyak kuchini aniqlaydigan minerallarning tarqalishini baholash imkonini beradi.

Suyak to‘qimalarining metabolizmi buzilishlarini aniqlashning biokimyoviy usullari:

- kalsiy-fosfor almashinuvi va kalsiyni tartibga soluvchi gormonlar;
- suyak to‘qimalarining shakllanishi va rezorbsiyasi belgilari.

Kalsiy-fosfor almashinuvi va kalsiyni tartibga soluvchi gormonlar ya’ni quyidagilar kiradi: fosfor va kalsiy kunlik chiqarilishini aniqlash shuningdek, kreatininga nisbatan kaltsiy,

ularning qondagi darajasini aniqlash, kalsitonin, paratiroid gormoni, D vitamin darajasini aniqlash.

Suyak to'qimalarining shakllanishi va rezorbsiyasi belgilari ya'ni gidroksid fosfataza (osteokalsin), kislota fosfataza, gidroksiprolin va boshqalar. Boshqa diagnostika usullarida bo'lgani kabi, vaqt o'tishi bilan olingan natijalarni solishtirish muhimdir.

Hozirda klinik amaliyotda uzoq muddatli ko'p markazli platsebo-nazoratli klinik tadqiqotlarda turli joylarda sinish xavfini kamaytirishda samarali ekanligi isbotlangan dorilar qo'llaniladi. Terapiyani tanlash o'ziga xos klinik holatga va individual dori-darmonlarni buyurishda, shuningdek, qabul qilishning afzal qilingan yo'liga: og'iz orqali, teri ostiga yoki tomir ichiga bog'liq.

Umurtqa pog'onasi osteoparozida quyidagi diagnostik belgilar muhim ahamiyatga ega:

- o'sishning pasayishi (tananing uzunligi qo'l oralig'idan 12 sm yoki undan ko'proq);
- umurtqa pog'onasining urish va palpatsiya paytida og'riq, orqa mushaklarning tonusini oshishi;
- egilish, ko'krak kifozining rivojlanishi va bel sohasida lordozning kuchayishi;
- umurtqa pog'onasi uzunligining qisqarishi tufayli yonbosh suyagii va pastki qovurg'alar orasidagi masofaning qisqarishi;
- qorinning yon tomonlarida teri burmalari paydo bo'lishi.

Skeletning boshqa suyaklarida umumiy og'riq sindromi kam uchraydi. Osteoporozning og'ir asorati son suyagi bo'yinining sinishi bo'lib, u yuqori o'lim, nogironlik va yuqori davolanish xarajatlariga sabab bo'ladi. Steroid (postmenopauzadagi ayol) osteoporozni ko'plab qovurg'a sinishi bilan tavsiflanadi.

Klinisistlar sekin rivojlanuvchi va o'tkir osteoporozni quyidagicha farqlaydilar.

▪ Sekin rivojlanuvchi osteoporoz uchun kasallikning boshlanishida o'tkir og'riqlar xos emas. Bu umurtqa pog'onasining asta-sekin o'zgaruvchan deformatsiyasi bilan bog'liq.

▪ O'tkir boshlanish lumbalgiya klinikasiga o'xshaydi va vertebra tanasini kompression sinishi bilan bog'liq, masalan, og'ir

yuk ko‘tarishdan keyin. Anamnezida qattiq og‘riq travma belgilarisiz 1-2 kun davom etadi. Umurtqa pog‘onasining osteoporozini davolash quyidagi holatlarda bo‘ladi. Osteoporoz uchun parhez quyidagi tamoyillarga rioya qilishi kerak:

1. Ratsionning asosiy vazifasi oziq-ovqatdan kalsiy va D vitamining yetarli darajada iste‘mol qilinishini ta‘minlashdir. Postmenopozal ayollar va 50 yoshdan oshgan erkaklar uchun oziq-ovqatdan kaltsiyning kunlik iste‘moli 1200-1500 mg bo‘lishi kerak.

2. Achitilgan sut mahsulotlari kabi eruvchan shakllarda kaltsiyi afzal ko‘rish tavsiya etiladi, garchi ba‘zi vegetarianlar (ayniqsa, vegetarianlar osteoporozning asosiy sababi bo‘lgan sut mahsulotlari, shuningdek, boshqa hayvonot mahsulotlari ekanligini ta‘kidlashadi.

3. Kaltsiy eng yaxshi 1 g yog‘ va 10 mg kaltsiy nisbatida so‘riladi. Bu nisbatning har ikki tomonga siljishi kaltsiyning so‘rilishini kamaytiradi.

4. Magniy, kaliy va fosfor kaltsiyning so‘rilishida muhim rol o‘ynaydi va bu mikroelementlar uchun dieta muvozanatli bo‘lishi kerak [16].

5. Kremniy, bor, rux, marganets, mis, vitamin C, D vitamini, E vitamini, K vitaminiga boy oziq-ovqatlarni yetarlicha iste‘mol qilish tavsiya etiladi.

Medikamentoz davolash. Osteoporoz uchun farmakoterapiyaga muhtoj bo‘lgan bemorlarni aniqlashda sinish xavfi yuqori bo‘lgan uchta guruh ajratiladi.

Anamnezida kam energiyali sinishlar bo‘lgan bemorlar;

- Kam suyak massasi bo‘lgan bemorlar;

- Osteoporotik yoriqlar xavfi yuqori bo‘lgan bemorlar.

Bularga keyingi 10 yil ichida asosiy osteoporotik sinishlar rivojlanish xavfi 20 % yoki son suyagi sinishi xavfi 3 % bo‘lgan bemorlar kiradi.

Bugungi kunda uzoq muddatli, ko‘p markazli, platsebo-nazorat ostida o‘tkazilgan klinik tadkikotlarda turli joylarda sinish xavfini kamaytirishda samarali ekanligi isbotlangan dori-darmonlar klinik amaliyotda qo‘llaniladi. Terapiyani tanlash o‘ziga xos klinik holatga va individual dori-darmonlarni buyurishda, shuningdek,

qabul qilishning afzal qilingan yoʻliga (ogʻiz orqali, teri ostiga yoki tomir ichiga) bogʻliq. Bunday preparatlarning teri osti shakli nafaqat yopishqoqlik qulayligi, balki iqtisodiy samaradorlik bilan ham afzalliklarga ega.

Osteoporozning oldini olishning umumiy qabul qilingan dori-darmon boʻlmagan usuli — bu turmush tarzini oʻzgartirish ya'ni jismoniy faollikni saqlash, quyoshga yetarli darajada ta'sir qilish, ayniqsa shimoliy mamlakatlarning keksa aholisi uchun, chekishni tashlash, spirtli ichimliklarni suiste'mol qilish uchun shifokorning bemor bilan o'quv ishlari uchun hamkorligiga alohida e'tibor qaratiladi.

- Toʻliq uyqu.
- Kunlik ratsiondagi kaltsiy miqdorini mintaqa uchun tavsiya etilgan darajada saqlash oʻrtacha, kuniga 1200-1500 mg.
- D vitamini darajasini saqlab turish. Yoshligida D vitaminining zarur miqdori biosintez bilan ta'minlanadi, ammo keksa va qarilikda D vitaminini qoʻshimcha yuborish kerak, bu uning organizmdagi kaltsiyning tarkibini normallashtiradi va soʻrilishini ta'minlaydi.

References

1. *C. Шваца, Ф. Спенсер.* Справочник по клиническому хирургу. Санкт-Петербург, Москва — Минск, 2000.
2. *M.X. Muxtorov, R.A. Bekmurodova, T.I. Rajabov.* Jarrohlik va reanimatsiya asoslari. T., «Meditsina», 1998.

CLINIC AND DIAGNOSTICS OF TRAUMATIC FRACTURES OF THE WRIST BONE AND OPTIMIZATION OF TREATMENT TACTICS

Kodirov T.¹, Odilov I.A.², Ahmedov B.O.³

¹*Kodirov Timur – Assistant,*

²*Odilov Ibrohim Abdulhamid ugli – Assistant,*

³*Ahmedov Bakhtiyor Odilovich – Assistant,*

ANDIJAN STATE MEDICAL INSTITUTE,

ANDIJAN, REPUBLIC OF UZBEKISTAN

Abstract: in this article, experiences are presented about the clinic and diagnosis of traumatic fractures of the wrist bone, as well as the processes of optimization of treatment tactics.

Keywords: wrist bone fracture, fracture, diagnosis, pathological fracture, bone fragment, tactics, optimization, spine, clinical syndrome, surgery.

UDC: 617.616

Bilak suyaklari sinishi haqida so'z ketarkan avvalo sinish jarayoniga xarakteristikasiga va sinish strukturasi batafsil to'xtalib o'tish maqsadga muvofiq bo'ladi.

Suyak sinishi — skeletning muayyan qismiga uning mustahkamligidan yuqori bo'lgan yuk tushishi natijasida suyak yaxlitligining to'liq yoki qisman buzilishi. Sinish jarohat tufayli, shuningdek suyak to'qimalarining mustahkamlik xususiyatlari o'zgarishi bilan kechadigan turli kasalliklar natijasida sodir bo'lishi mumkin. Holatning jiddiyligi shikastlangan suyaklar soni va jarohat ko'lami bilan belgilanadi. Bir qancha naysimon suyaklarning sinishi ko'p qon yo'qotish va og'riqli shok rivojlanishiga olib keladi. Bundan tashqari, bemorlar bunday jarohatlardan so'ng sekin tiklanishadi, tuzalish bir necha oyga cho'zilishi mumkin.

Yuzaga kelish sababiga ko'ra

- **Travmatik** — tashqi ta'sir tomonidan chaqirilgan;
- **Patologik** — suyakning ba'zi patologik jarayonlarda (masalan, tuberkulyoz, shish va boshqalar) minimal tashqi ta'sir ta'siri natijasida ham sinishi.

Shikastlanish og'irligiga ko'ra

- **To'liq:**
- Siljishsiz (masalan, periost ostida);
- Bo'laklarning siljishi bilan.
- **Noto'liq** — yoriqlar va qisman sinish.

Sinish shakli va yo'nalishi bo'yicha

- **Ko'ndalang** — sinish chizig'i naysimon suyak o'qiga shartli perpendikulyar;

- **Uzunasiga** — sinish chizig'i naysimon suyak o'qiga shartli parallel;
- **Qiyshiq** — sinish chizig'i naysimon suyak o'qiga nisbatan o'tkir burchak ostida o'tadi;
- **Vintsimon** — suyak parchalarining o'zining normal joylashuviga nisbatan aylanishi;
- **Parchalangan** — bitta aniq sinish chizig'i yo'q, shikastlangan joydagi suyak alohida bo'laklarga bo'linishi;
- **Ponasion** — odatda umurtqa sinishida qayd qilinadi, bunda bir suyak boshqa suyakka bosilib kirib qoladi va pona shaklidagi deformatsiyani hosil qiladi;
- **Sanchilgan** — suyak bo'laklari naysimon suyak o'qi bo'ylab proksimal slijydi yoki asosiy naysimon suyak sirtidan tashqarida joylashadi;
- **Kompression** — suyak parchalari juda kichik, aniq, bitta sinish chizig'i yo'q.

Terining yaxlitligiga ko'ra

- **Yopiq** — sinish joyidagi to'qimalarning jarohatlanishi qayd etilmaydi va jarohat tashqi muhit bilan aloqa qilmaydi. Yakka yoki bir nechta kabi guruhlariga bo'linadi;
- **Ochiq** — suyak sinishida (o'qotar qurol bilan yaralangan yoki yaralanmagan) yumshoq to'qimalarning shikastlanishi va tashqi muhit bilan aloqa qilishi qayd qilinadi.

Sinishing lokalizatsiyasiga ko'ra

Naysimon suyaklar doirasida quyidagilar ajratiladi:

- Diafiza;
- Epifiza;
- Metafiza.

Asoratlariga ko'ra

- Asoratlangan:
- Travmatik shok;
- Ichki a'zolarining shikastlanishi;
- Qon ketishi;
- Yog'li emboliya;
- Jarohatga infektsiya tushishi, osteomielit, sepsis.
- Asoratlanmagan. Bilak suyaklarining sinishlari yopiq sinishlarning umumiy soniga 30 % gacha qismni tashkil qiladi va

og'ir holda murakkab shikastlanishlar qatoriga kiradi. Bilak suyagi boshchasi va bo'ynining sinishi bilvosita travma oqibatida yozilgan qo'l bilan yiqilganda paydo bo'ladi. Buning natijasida bilak suyagining boshchasi yelkaning boshsimon do'ngiga kat ga kuch bilan uriladi.

Tashxis. Tirsak chukurchasi ko'zdan kechirilganda yelka-bilak birikmasi proektsiyasi ustida yaqqolrok ko'rinadigan shish aniqlanadi. Paypaslab ko'rishda harakat paytida kuchayadigan mahalliy og'riq, bo'ladi. Aktiv xarakterlar cheklangan va og'riqli (qo'lni yozish keskin chegaralangan, bilakning tashqariga rotatsiyasi kattik okrik bilan o'tadi). Bilak suyagining boshchasi pronatsiya va supinatsiyada katnashmaydi. Shu suyak o'ki bo'ylab og'irlik tushishi og'riq. beradi. Ikki proektsiyadagi reshtenofafiya tashxisni aniq.ashtirishga yordam beradi.

Davolash. Bilak suyagining boshchasi va bo'ynining surilmagan yoki oz-moz surilgan va boshchasi 20°ga organ sinishlarida tirsak bo'g'imida 90-100° burchak ostida buqilgan qo'lga kaft-barmoq. birikma-laridan yelkaning o'rtasigacha gipsli bog'lam ko'yiladi. Bunda bilakka supinatsiya bilan pronatsiya orasidagi o'rtacha xolat beriladi. Mexnat krbiliyati 7-8 xafta o'tgach tiklanadi. Bo'ynining surilib va boshchasining 20°dan ko'pga organ sinishlarida bir momenta qo'l repozitsiya qilinadi. Bo'laklar repozitsiyasi yaxshi natija bermasa, jarroxlik usuli qo'llaniladi: bilak suyagi boshchasini ochiqjoyiga tushuriladi va kegay yoki suyak shtifti bilan transartikulyar fiksatsiya qilinadi. Agar bilak suyagi boshchasi joyiga tushmasa, uni olib tashlanadi. Boshcha, shuningdek maydalangan sinishda va uning bo'g'im qismidan 1/3 ulushi surilgan boshcha sinishida xam olib tashlanadi. Jarroxlik usulidan so'ng bilakning pronatsiya bilan supinatsiya orasidagi o'rta xolati tirsak bo'g'imi 90°gacha buqilib, 10-15 kunga gipsli bog'lam qo'yiladi. So'ngra bo'g'imda asta-sekin xarakterlar qilina boshlanadi va 1,5-2 oy o'tgach tiklanadi. Bilak suyagi diafizining cheklangan sinishi, odatda, bevosita shikast ta'siri ostida paydo bo'ladi. Sinish chizish aksariyat ko'ndalang joylashadi. Bilak suyagi xar kanday satxidan sinishi mumkin, biroq eng ko'p — bilakning pastki va o'rta

chegarasida, bilak uchdan bir qismining o'rtta va yuqori qismida joylashgan bo'ladi.

Tashxis. Bilak suyagi tirsak suyagidan ichkarida joylashgan, shuning uchun tirsak suyagi butun bo'lganda bilak suyagining cheklangan sinishini ko'pincha aniqlash qiyin. Singan joy ko'zdan kechirilganda bo'laklarning surilishi va yumshoq to'qimalar shishi tufayli deformatsiya ko'rinadi. Bilak suyagi paypaslanganda singan soxada bosilganda kuchayadigan maxalliy og'riq aniqlanadi. Bilak o'qi bo'ylab og'irlik singan soxada ogriydi. Bilak suyaklariga tushadigan yaqinlashtiruvchi yuk xam singan joyda og'riqning kuchayishiga olib keladi. Bilak suyagi diafizining sinishi uchun bilakning aktiv pronatsion va supinatsion xarakterlarining yo'qligiga xos.

Bilakning passiv rotatsion xarakatlari keskin og'riqli. Bilak suyagi boshchasi bunda xarakatsiz qolishi mumkin. Tashxisni aniqlash uchun tirsak va bilak-kaft bo'g'imlarini kamrab olgan ikki proektsiyali rentgenografiya qilish zarur.

Davolash. Bilak suyagi diafizining uchdan bir yuqori va o'rtta qismidagi bo'laklari surilmagan cheklangan sinishlarini barmoqlar asosidan to yelkaning uchdan bir yuqori qismigacha gipsli bog'lamda davolanadi. Tirsak bo'g'imida 90°gacha bukilgan bilak supinatsiya holatida fiksatsiya qilinadi. Bilak suyagi diafizi uchdan bir pastki khismida singanda gipsli bog'lamni yelkaning uchdan bir pastki qismigacha qo'yiladi. Bilak bunda pronatsiya va supinatsiya o'rtasidagi o'rtta xolatda fiksatsiya qilinadi. Immobilizatsiya 8-10 xafta mobaynida o'tkaziladi. Mehnat qobiliyati 10-11 xafta o'tgach tiklanadi.

Bilak suyagi diafizi bo'laklari surilib singan xollarda ularni repozitsiya qilish zarur. Bilak suyagi diafizi dumalok pronator yopishgan joydan yuqorida joylashadigan sinishda surilgan bo'laklar bilakda to'liq supinatsiya bo'lganda va u o'z o'qi atrofida traktsiya qilinganda o'z joyiga tushadi. Sinish dumaloq pronator yopishgan joydan pastda bo'lganda repozitsiya bilakning xolati pronatsiya bilan supinatsiya o'rtasidagi o'rtta xolatda repozitsiya o'gkz qilinishi kerak. Bilak suyagi diafizi bo'laklari surilgan uchdan bir pastki qismi sinishlarida repozitsiyani bilakning to'liq supinatsiyasi, abduktsiyasi, ekstenziyasi va 1-

barmoqdan qattiq tortish bilan o'tkaziladi. Bunda jarrox deformatsiya soxasini bosib, surilishni bartaraf qilishga imkon beradi. So'ngra, cho'zish va bilak rotatsiyasi saqlangan xolda yelkaga va bilakka uning supinatsiyasini saqlagan xolda gipsli bog'lam qo'yiladi.

References

1. *Г.Р. Мак Летчи.* Оксфордский журнал клинической хирургии М., “Медицина”, 1999.
2. *С. Шваца, Ф. Спенсер.* Справочник по клиническому хирургу. Санкт-Петербург, Москва — Минск., 2000.
3. *В.И. Стручков.* Общая хирургия. М., “Медицина”, 1983.

НАУЧНОЕ ИЗДАНИЕ

**ИЗДАТЕЛЬСТВО
«НАУЧНЫЕ ПУБЛИКАЦИИ»**

**АДРЕС РЕДАКЦИИ:
153000, РФ, ИВАНОВСКАЯ ОБЛ., Г. ИВАНОВО,
УЛ. КРАСНОЙ АРМИИ, Д. 20, 3 ЭТАЖ, КАБ. 3-3,
ТЕЛ.: +7 (915) 814-09-51.**

**[HTTPS://SCIENTIFICPUBLICATION.RU](https://scientificpublication.ru)
EMAIL: TEL9203579334@YANDEX.RU**

**ИЗДАТЕЛЬ:
ООО «ОЛИМП»
153002, РФ, ИВАНОВСКАЯ ОБЛ., Г. ИВАНОВО, УЛ. ЖИДЕЛЕВА, Д. 19
УЧРЕДИТЕЛЬ: ВАЛЬЦЕВ СЕРГЕЙ ВИТАЛЬЕВИЧ**



ИЗДАТЕЛЬСТВО «НАУЧНЫЕ ПУБЛИКАЦИИ»
HTTPS://SCIENTIFICPUBLICATIONS.RU
EMAIL: INFO@SCIENTIFICPUBLICATIONS.RU

 **РОСКОНАДЗОР**
СВИДЕТЕЛЬСТВО ЭЛ № ФС 77–65699



INTERNATIONAL STANDARD
SERIAL NUMBER 2542-081X

Российская
книжная палата
ТАСС

 Google™
scholar

 **РОССИЙСКИЙ
ИМПАКТ-ФАКТОР**
IMPACT-FACTOR.RU



Вы можете свободно делиться (обмениваться) — копировать и распространять материалы и создавать новое, опираясь на эти материалы, с ОБЯЗАТЕЛЬНЫМ указанием авторства. Подробнее о правилах цитирования: <https://creativecommons.org/licenses/by-sa/4.0/deed.ru>

ЦЕНА СВОБОДНАЯ