

CROSS PLATFORM MOBILE DEVELOPMENT



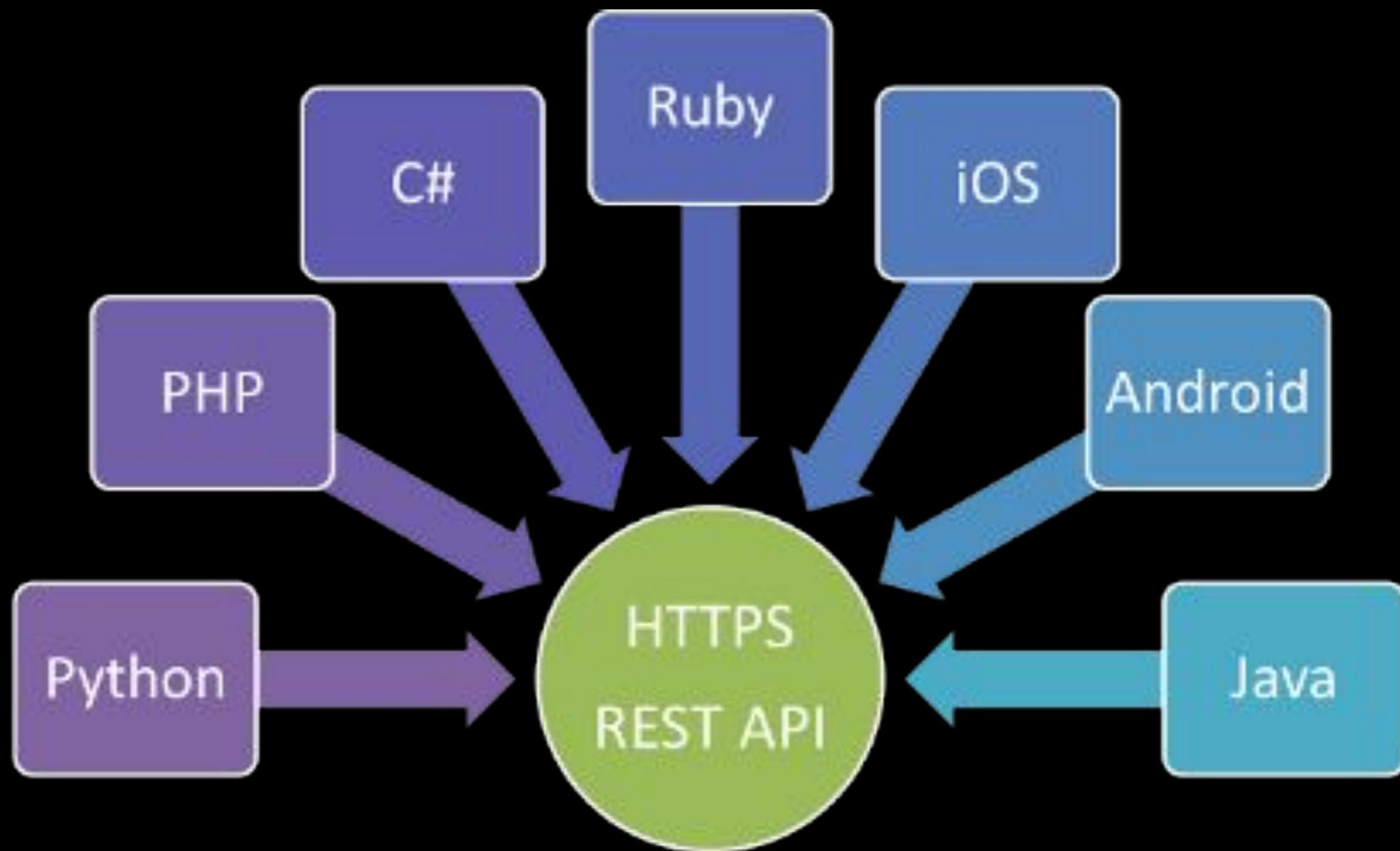
REST API



REST - что это?

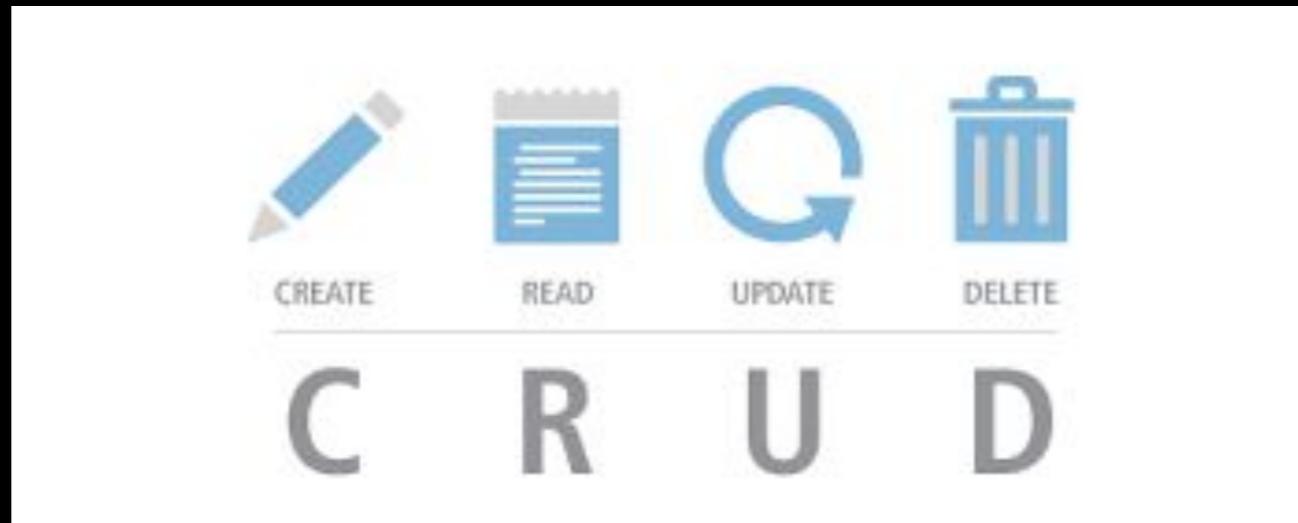
- Representational State Transfer — «передача репрезентативного состояния». Метод взаимодействия компонентов распределённого приложения в сети Интернет, при котором вызов удаленной процедуры представляет собой обычный HTTP-запрос (обычно GET или POST; такой запрос называют REST-запрос), а необходимые данные передаются в качестве параметров запроса.
- Этот способ является альтернативой более сложным методам, таким как SOAP, CORBA и RPC. В широком смысле REST означает концепцию построения распределённого приложения, при которой компоненты взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине.





Всем нужен сервер

REpresentational State Transfer === CRUD



	Structured Query Language	HTTP	Data Distribution Service
Create	INSERT	PUT / POST	write
Read (Retrieve)	SELECT	GET	read / take
Update (Modify)	UPDATE	PUT / PATCH	write
Delete (Destroy)	DELETE	DELETE	dispose

Что внутри в двух словах

<http://api/>

{название
ресурса}/ {уникальный
идентификатор}

?параметр=значение

ЧТО ВНУТРИ В ДВУХ СЛОВАХ

- **GET**

- Отдай ресурс
- Отдай событие

- **POST / PUT**

- Создай ресурс
- Создай событие

Что внутри в двух словах

- **PATCH / PUT**
 - Обнови ресурс
 - Создай если её нет
- **DELETE**
 - Удали ресурс

ЧТО ВНУТРИ В ДВУХ СЛОВАХ

- **HEAD**

- Отдай ресурс
- Отдай событие
- **Но всё это без содержимого, только заголовки**

- **CONNECT**

- Создай подключение TCP/IP tunnel или HTTPS

- **OPTIONS**

- Покажи доступные мне методы

- **TRACE**

- Верни мне что я тебе прислал



REST - не стандарт



REST - не стандарт

1. Все является ресурсами с уникальным идентификатором (URL)
2. Все операции - это ресурсы или сущности
3. Любой запрос это CRUD

GET

- **GET /resource**

Отдать данные так как задумано сервером

- **GET /resource/{id}**

Уникальный идентификатор это ID, hash, slug, username - что-то уникальное что является идентификатором

- **GET /resource/{id_1},{id_2},{id_3}**

Клиент хочет несколько ресурсов, отдай их массивом

GET

- **GET /tavern/Admiral_Benbow/checkin**
Посмотреть все чекины данном месте
- **GET /pirate/Billy_Bones/checkin**
Посмотреть все чекины пользователя
- **GET**
/pirate/Billy_Bones/checkin/Admiral_Benbow
Посмотреть есть ли чекин пользователя в данной местности вернув запись

GET

- **<http://api/user/12345>**
я знаю сколько у тебя пользователей
- **<http://api/user/@HanSolo>**
я знаю имя пользователя или логин или email

GET

<http://api/tavern/4>

```
{  
  "status": "ok",  
  "data": [  
    {  
      "id": "abcd-efgh",  
      "name": "Tavern 'Dancing fairy'"  
    }  
  ],  
  "next": "http://api/tavern/?offset=5",  
  "prev": "http://api/tavern/?offset=3",  
}
```



GET

- **DELETE /account/{id}**
удаляем пользователя
- **DELETE /account/{id_1},{id_2},{id_3}**
удаляем много пользователей
- **DELETE /account**
удалить все записи ресурса (!!!)
- **DELETE /account/{id}/avatar**
удаляем аватарку
- **DELETE /account/{id}/avatar/***
удалить все аватарки

Вложенные данные

```
[  
  {"name": "Jay", "phrase": [1, 2]},  
  {"name": "Silent Bob", "phrase": ["..."]},  
  {"name": "Metatron", "phrase":  
    {"5": "Behold, the Metatron!",  
     "6": "Metatron acts as the voice of God."}},  
  {"name": "Azrael",  
   "phrase": ["http://api/phrase/7",  
             "http://api/phrase/8"]}  
]
```



Разбиение на страницы

```
{  
  "pagination": {  
    "total": 1003,  
    "per_page": 10,  
    "total_pages": 100,  
    "current_page": 3,  
    "next_page": 4,  
    "prev_page": 2,  
    "next_url": "http://api/resource/:page/4",  
    "prev_url": "http://api/resource/:page/2"  
  }  
}
```

За пределами CRUD

Версионирование API

- **URL**

`https://api/v1.0.5/resource`

`https://api/v2.1.9/resource/123456`

- **Параметр**

`https://api/resource?v=3.2.6`

- **Заголовки**

`Accept: application/json;version=v1.2.4`

`Accept: application/json;api=NameOfMyApi;version=v1.4.6`



ПОИСК

POST	/:search	Новый поиск
GET	/:search	Что недавно искали
GET	/:search/{id}	Идентификатор поиска
DELETE	/:search/{id}	Удалить кеш

http://www.domain.my/:search/?what=articles&text=Тolkien*&request_id=123456

```
202 Accepted
{
  "incremental_call": 60000,
  "left": 53,
  "per_day": 200
}
```



ОШИБКИ

```
{  
  "status": "Forbidden",  
  "http_code": 403,  
  "code": "SESSION_EXPIRED",  
  "type": "OAuthException",  
  "message": "Session has expired"  
}
```

```
{  
  
  "status": "OK",  
  "http_code": 404,  
  "code": «DEPRECATED_API_VERSION»,  
  "type": "APIException",  
  "message": "This API version is no longer supported"  
}
```

Документирование REST API

Документирование REST API

1. Код постоянно меняется
2. Время на документацию не выделяют
3. Документация не успевает за изменениями
4. Никто не умеет писать документацию
5. Когда начинать вести документацию?
6. Не хочу писать документацию

Service-Oriented Architecture

- HATEOAS
- “Hypermedia as the Engine of Application State”
- JSON Schema
- JSON Hyper-Schema

GET OPTIONS <http://api/user>

```
{
  "title": "User",
  "type": "object",
  "description": "Service User",
  "properties": {
    "firstName": {"type": "string"},
    "age": {
      "description": "Age in years",
      "type": "integer",
      "minimum": 0
    }
  },
  "required": ["firstName", "age"]
}
```

GET OPTIONS

<http://api/product>

HTTP/1.1 200 OK

Host: api.example.com

Content-Type: application/json

Cache-Control: no-cache, private

X-Methods: GET,POST,PUT,PATCH,DELETE

Apache Cordova

Apache Cordova — это платформа разработки мобильных приложений с открытым исходным кодом. Она позволяет использовать стандартные веб-технологии для кросс платформенной разработки, избегая родного языка разработки для каждой из мобильных платформ.

Приложения выполняются внутри обертки нацеленной на каждую платформу и полагаются на стандартные API для доступа к датчикам устройства, данным и состоянию сети.

ОСНОВНЫЕ ПРИНЦИПЫ

- Разработка ведется с использованием стека технологий HTML, CSS, Javascript
- Компиляция в нативное приложение возможно только при наличии необходимого SDK
- Имеет большой набор плагинов для использования «системного javascript»

Cordova: Плюсы

- **Удобная разработка:**
Нет необходимости постоянно компилировать проект, разработка ведется прямо в браузере
- **Наиболее адекватный порог вхождения**
- **Плагины**
- **Отладка уже запущенного приложения
прямо из браузера**

Cordova: МИНУСЫ

- Нативное приложение представляет собой форму с «натянутым» Web-view, что означает всю работу внутри одного Activity
- Чуть медленная производительность по сравнению с нативным приложением
- Функциональность зависит от текущей ОС и функции браузера
- Подписка на PUSH уведомления при выключенной программе (на стадии доработки)

IONIC FRAMEWORK

Что такое IONIC FRAMEWORK

- Набор дополнительных инструментов для Apache Cordova
- Стилевая функциональность для удобной верстки приложения
- Набор готовых модулей, альтернативных нативным (например, диалоги или InfinityScroll)
- В основе лежит Angular JS