

Skillbox

REST API

Шулаев Андрей

Senior Software Engineer at EPAM Systems

REST API: что это и зачем?

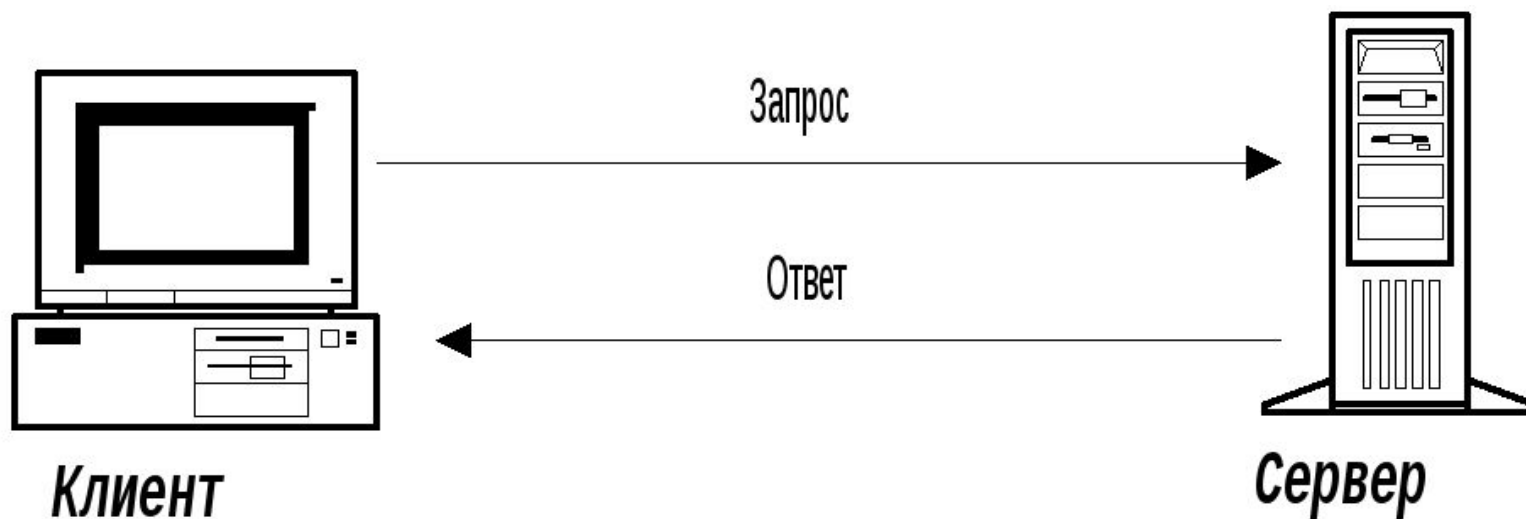
Skillbox

Рой Филдинг



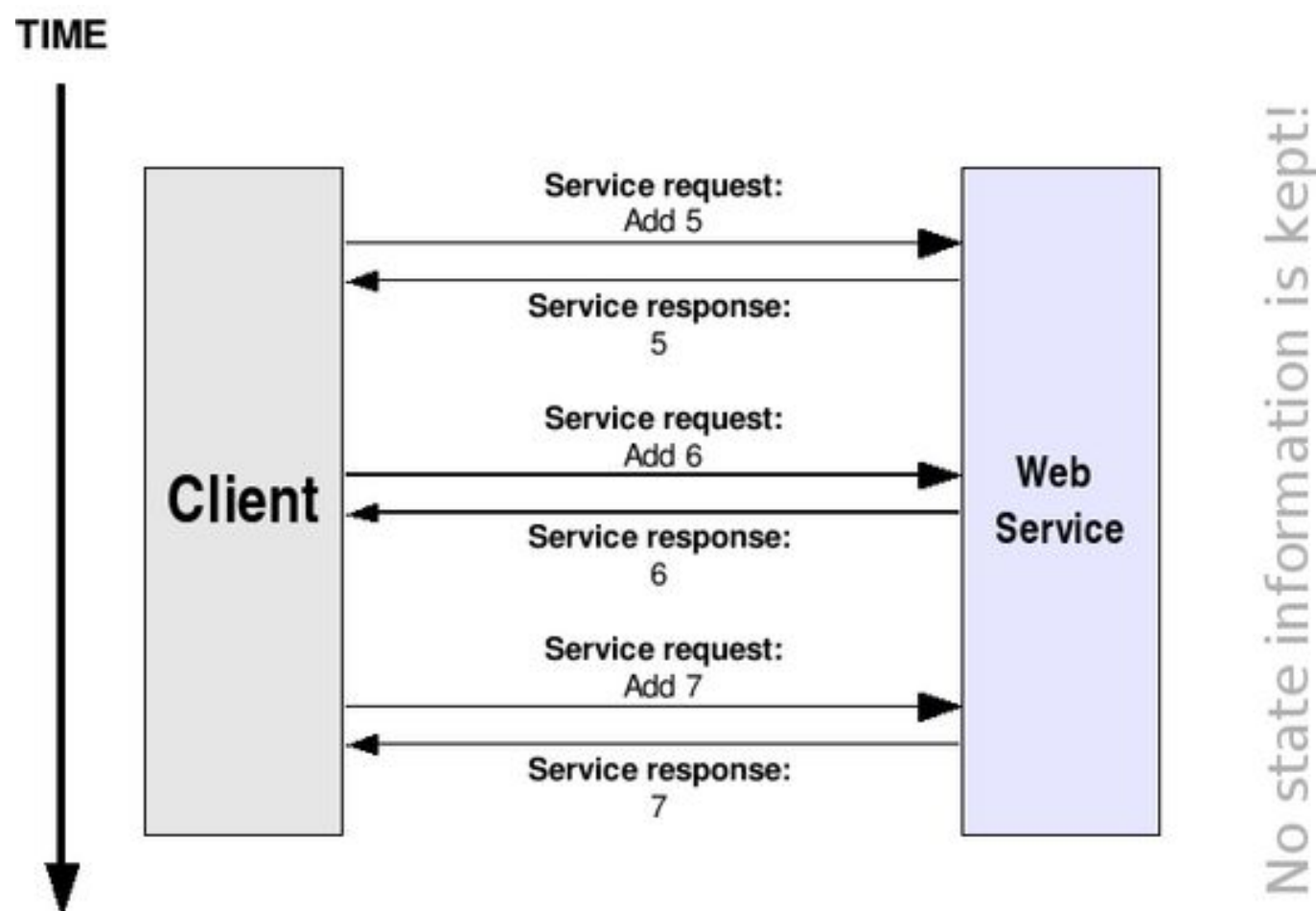
Требования к REST-архитектуре

1. Клиент-серверная архитектура



Требования к REST-архитектуре

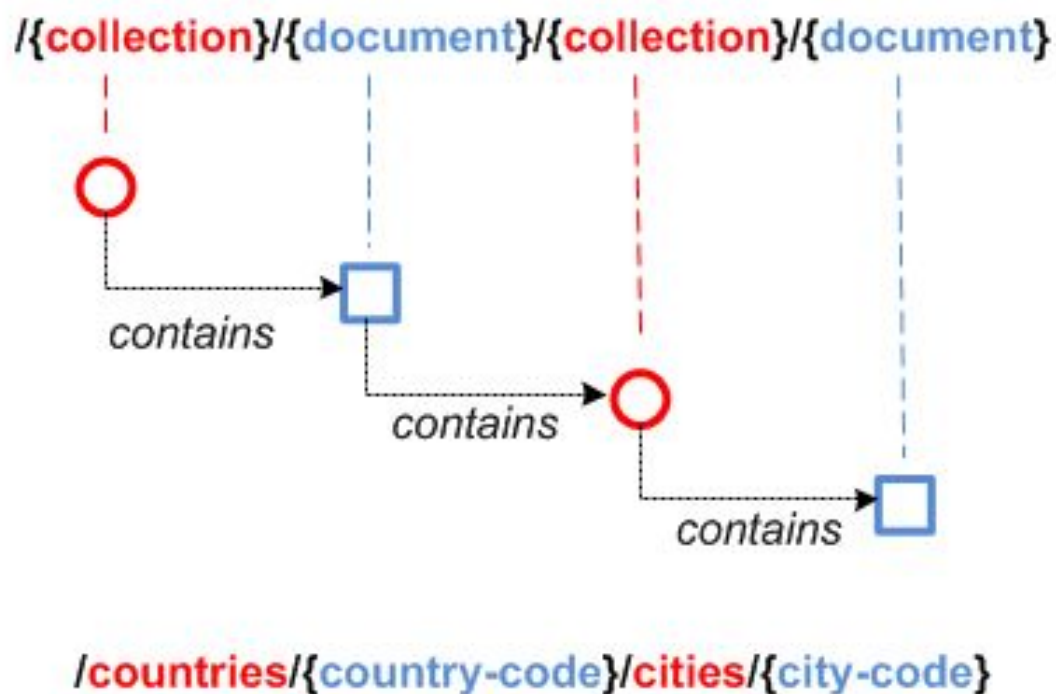
2. Отсутствие состояния (stateless)



Требования к REST-архитектуре

3. Единообразие интерфейса

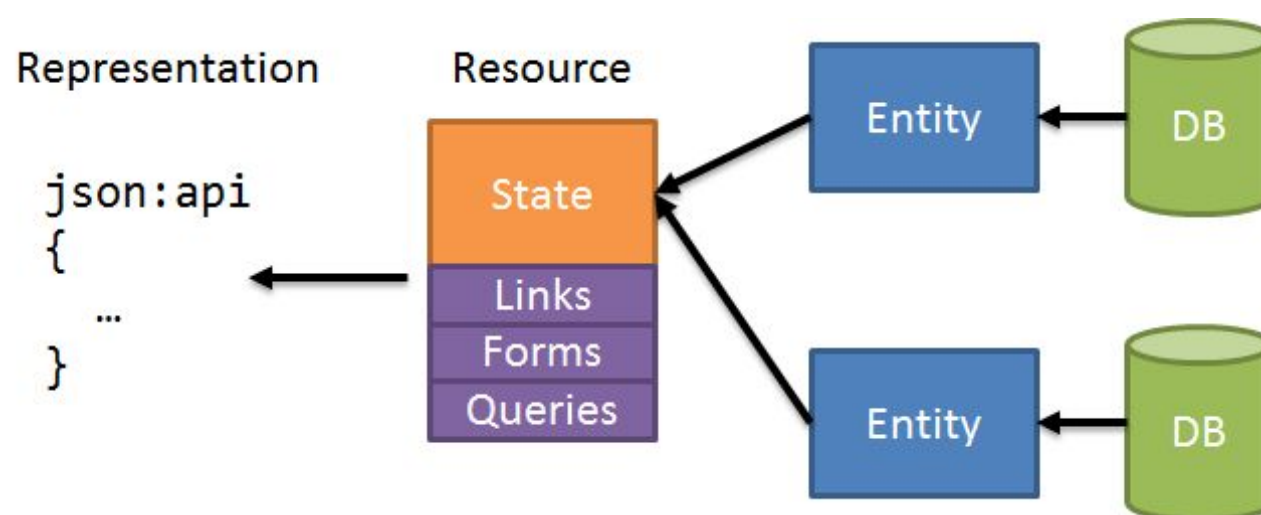
- Идентификация ресурсов



Требования к REST-архитектуре

3. Единообразие интерфейса

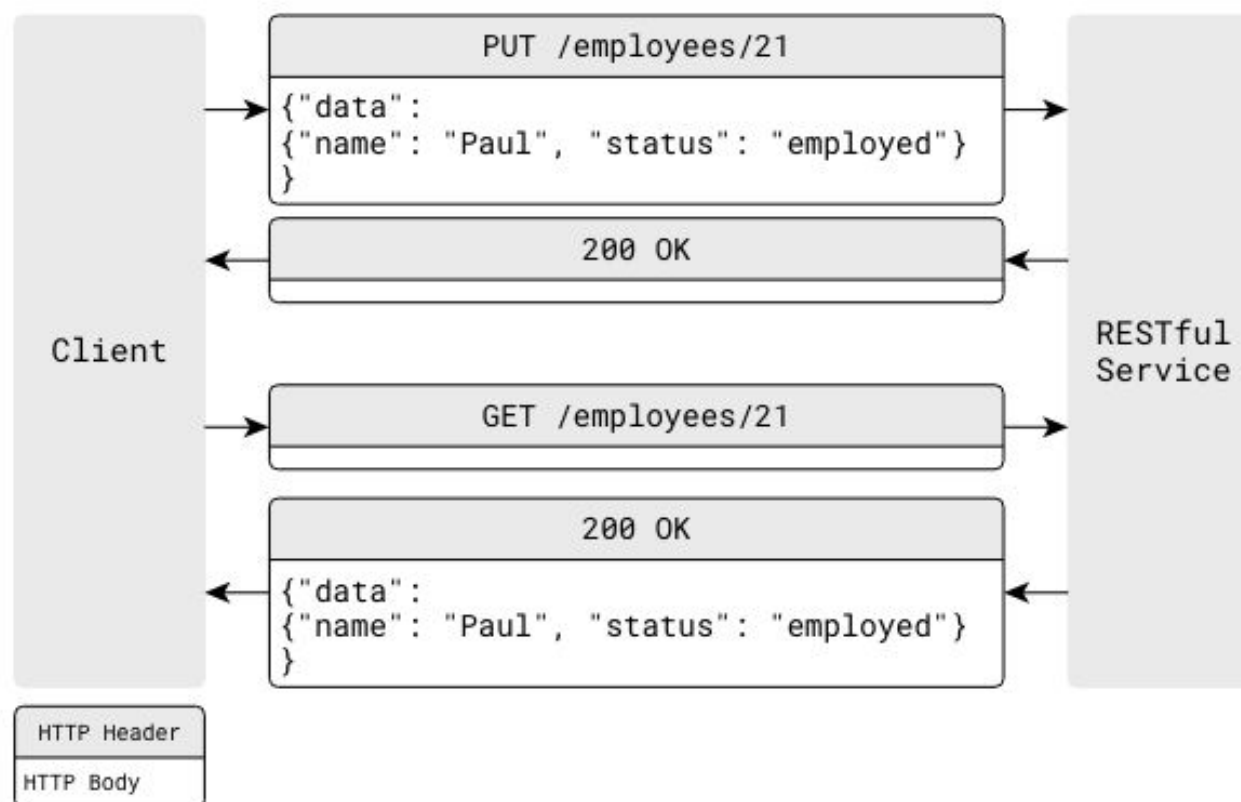
- Управление ресурсами через представление



Требования к REST-архитектуре

3. Единообразие интерфейса

- Самодостаточные сообщения



Требования к REST-архитектуре

3. Единообразие интерфейса

- Самодостаточные сообщения

```
GET / HTTP/1.1  
Host: www.example.com
```

Требования к REST-архитектуре

3. Единообразие интерфейса

- Самодостаточные сообщения

```
HTTP / 1.1 200 OK
Content-Type: text/html
<!DOCTYPE html>
Hello, User!
```

Требования к REST-архитектуре

3. Единообразие интерфейса

- Гипермедиа (HATEOAS)

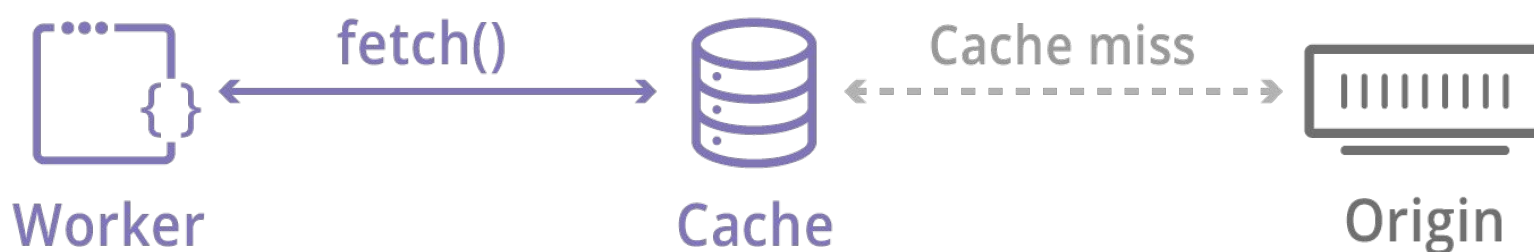
```
{
  "id": 1,
  "firstname": "Sergio",
  "lastname": "Leone",
  "year": 1929,
  "_links": {
    "self": {
      "href": "http://localhost:8080/directors/1"
    },
    "director_movies": {
      "href": "http://localhost:8080/directors/1/movies"
    },
    "directors": {
      "href": "http://localhost:8080/directors"
    }
  }
}
```

Data content

Hypermedia

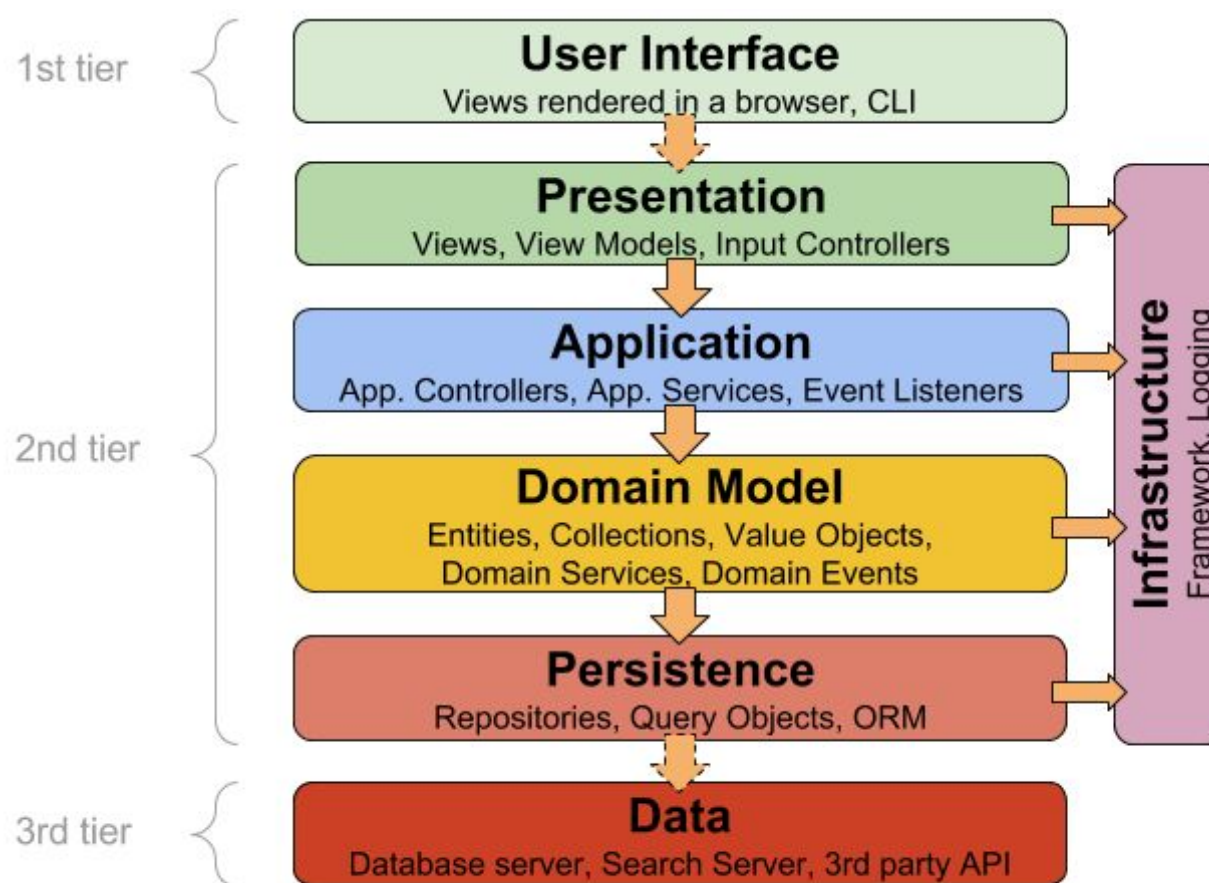
Требования к REST-архитектуре

4. Кэширование



Требования к REST-архитектуре

5. Система слоёв



Требования к REST-архитектуре

6. Код по требованию

```
▼<script type="text/javascript">
```

```
try { (function injectPageScriptAPI(scriptName, shouldOverrideWebSocket, shouldOverrideWebRTC,
isInjected) { 'use strict'; /** * If script have been injected into a frame via contentWindow then we
can simply take the copy of messageChannel left for us by parent document * Otherwise creates new message
channel that sends a message to the content-script to check if request should be allowed or not. */ var
messageChannel = isInjected ? window[scriptName] : (function () { // Save original postMessage and
addEventListener functions to prevent webpage from tampering both. var postMessage = window.postMessage;
var addEventListener = window.addEventListener; // Current request ID (incremented every time we send a
new message) var currentRequestId = 0; var requestsMap = {}; /** * Handles messages sent from the
content script back to the page script. * * @param event Event with necessary data */ var
onMessageReceived = function (event) { if (!event.data || !event.data.direction || event.data.direction
!== "to-page-script@abu") { return; } var requestData = requestsMap[event.data.requestId]; if
(requestData) { var wrapper = requestData.wrapper; requestData.onResponseReceived(wrapper,
event.data.block); delete requestsMap[event.data.requestId]; } }; /** * @param url The URL to which
wrapped object is willing to connect * @param requestType Request type ( WEBSOCKET or WEBRTC) * @param
wrapper WebSocket wrapper instance * @param onResponseReceived Called when response is received */ var
sendMessage = function (url, requestType, wrapper, onResponseReceived) { if (currentRequestId === 0) {
// Subscribe to response when this method is called for the first time addEventListener.call(window,
"message", onMessageReceived, false); } var requestId = ++currentRequestId; requestsMap[requestId] = {
```

Skillbox

**Спасибо
за внимание!**