

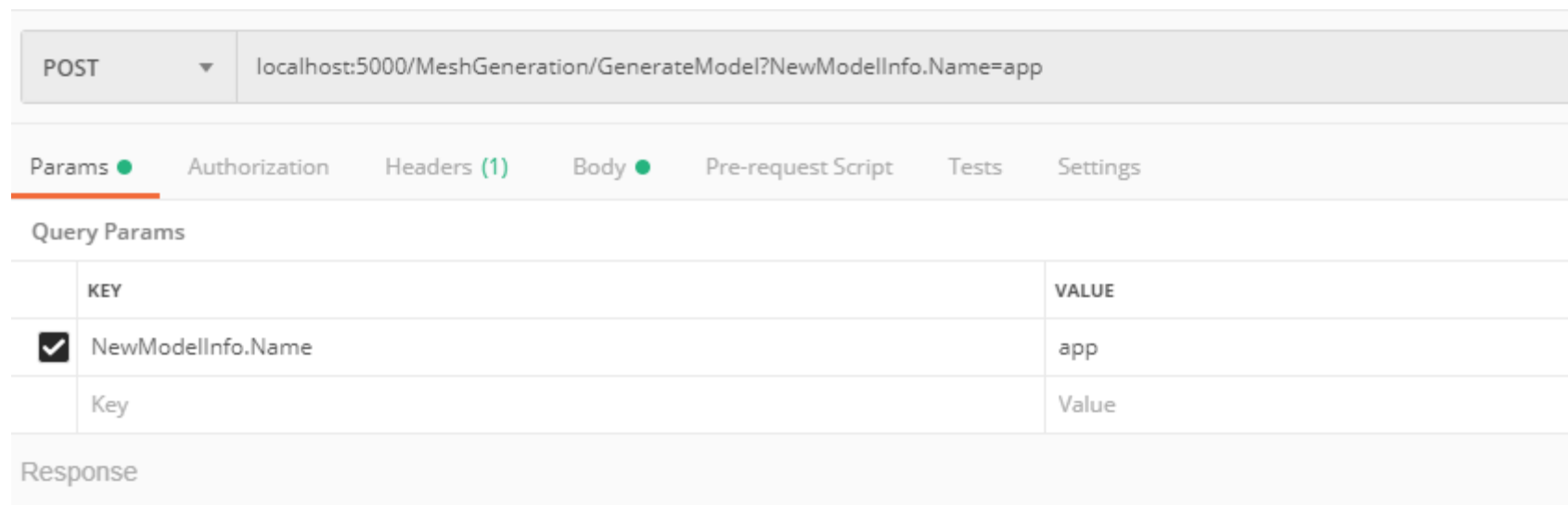
Лекция 8

План

- Инструменты
- Rails
- Паттерны
- YAML
- Роутинг

Инструменты для разработки API

- Postman
- Swagger

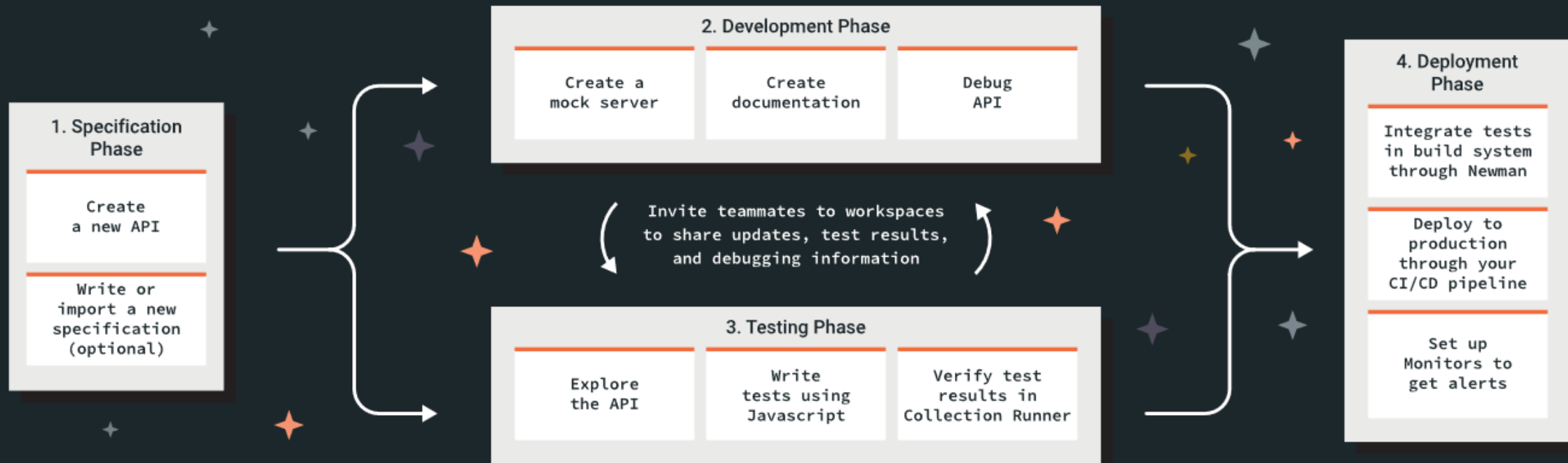


The screenshot displays the Postman interface for configuring a POST request. The URL is set to `localhost:5000/MeshGeneration/GenerateModel?NewModelInfo.Name=app`. The 'Params' tab is selected, showing a table of query parameters.

KEY	VALUE
<input checked="" type="checkbox"/> NewModelInfo.Name	app
Key	Value

Below the table, the 'Response' section is visible but empty.

Процесс разработки



Swagger

Schemes

HTTPS

Authorize



pet Everything about your Pets

Find out more: <http://swagger.io>



GET /pet/{petId} Find pet by ID



POST /pet/{petId} Updates a pet in the store with form data



DELETE /pet/{petId} Deletes a pet



POST /pet/{petId}/uploadImage uploads an image



POST /pet Add a new pet to the store



PUT /pet Update an existing pet



GET /pet/findByStatus Finds Pets by status



GET /pet/findByTags Finds Pets by tags



Визуальное представление

PUT /pet Update an existing pet 🔒

Parameters Try it out

Name	Description
body * required object (body)	Pet object that needs to be added to the store

Example Value | Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type

application/json ▾

ТЕКСТОВЫЙ формат

paths:

/inventory:

get:

tags:

- developers

summary: searches inventory

operationId: searchInventory

description: |

By passing in the appropriate options, you can search for available inventory in the system

produces:

- application/json

parameters:

parameters:

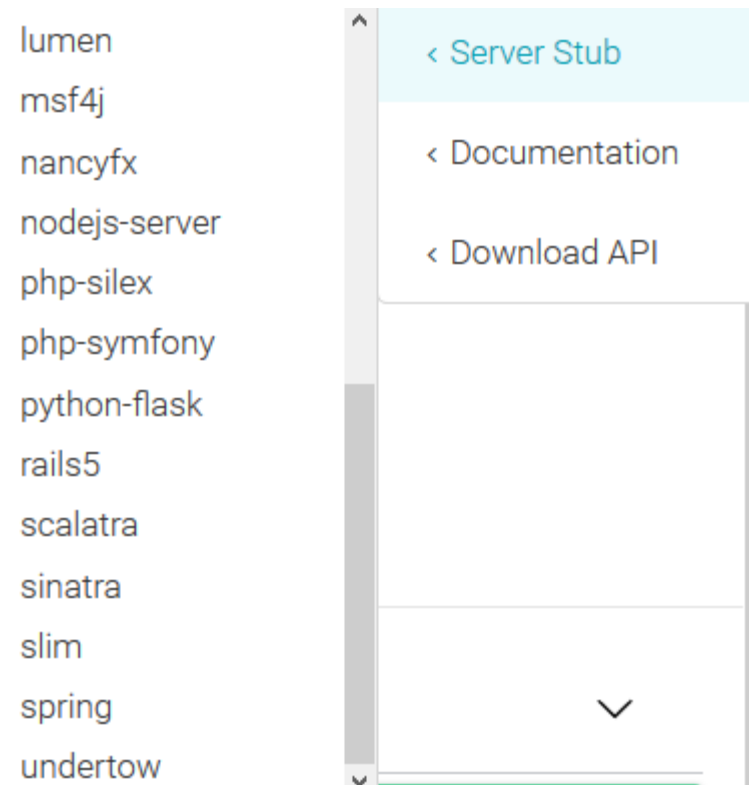
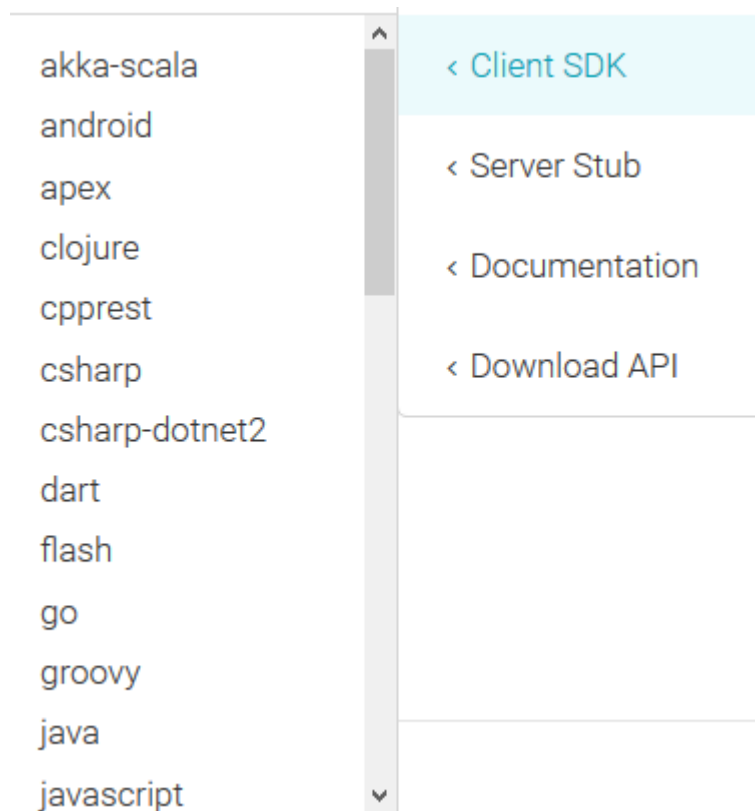
- in: query

name: searchString

Зачем это нужно?

- Унификация интерфейсов
- Готовая документация, еще и с онлайн-формой
- Упрощение тестирования
- Кодогенерация!

Кодогенерация



Swagger + Sinatra

```
require './lib/swaggering'
```

```
class MyApp < Swaggering  
  self.configure do |config|  
    config.api_version = '1.0.0'  
  end  
end
```

```
Dir["./api/*.rb"].each { |file|  
  require file  
}
```

* api.rb

```
MyApp.add_route('POST', '/example/1.0.0/inventory', {
  "resourcePath" => "/Admins",
  "summary" => "adds an inventory item",
  "nickname" => "add_inventory",
  "responseClass" => "void",
  "endpoint" => "/inventory",
  "notes" => "Adds an item to the system",
  "parameters" => [
    {
      "name" => "body",
      "description" => "Inventory item to add",
      "dataType" => "InventoryItem",
      "paramType" => "body",
    }
  ]}) do
  cross_origin
  # the guts live here

  {"message" => "yes, it worked"}.to_json
end
```

Ruby on Rails

DRY

Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.

«Каждый объект знаний должен быть присутствовать в системы единожды, однозначно и явно»

Convention Over Configuration

- Мы договорились, что это работает так.
- Если вам не нравится, это ваша проблема, можете настроить как угодно, но мы помогать не станем

Rails – это фреймворк

- Каркасный подход
- Холодные и горячие точки
- Абстракции и реализации

Паттерны web-приложений

- Model-View-Controller
- Model-View-Presenter
- Model-View-View Model

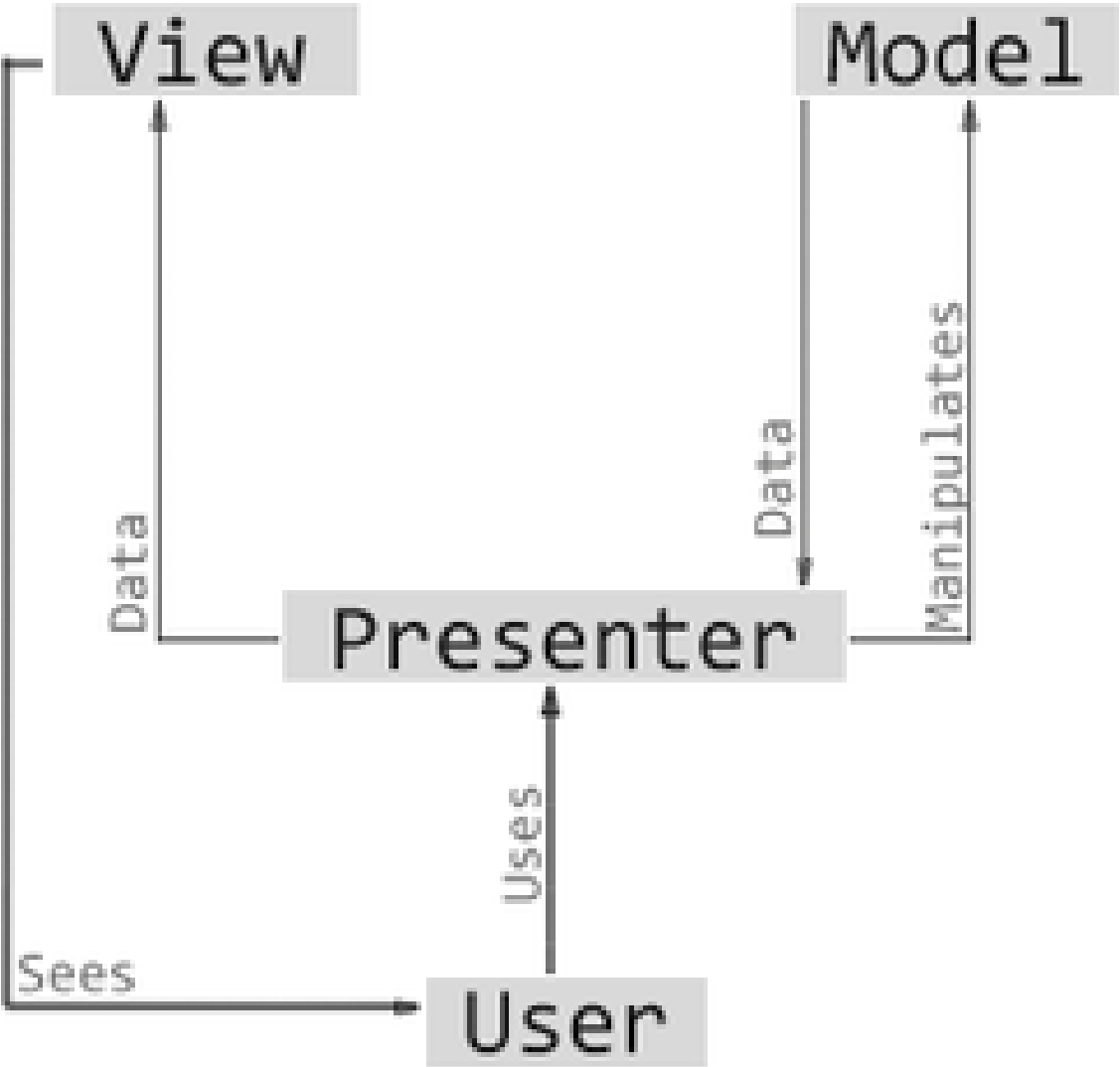
Model

- Слой бизнес-логики
- Passive Model
- Active Model /
Active Record

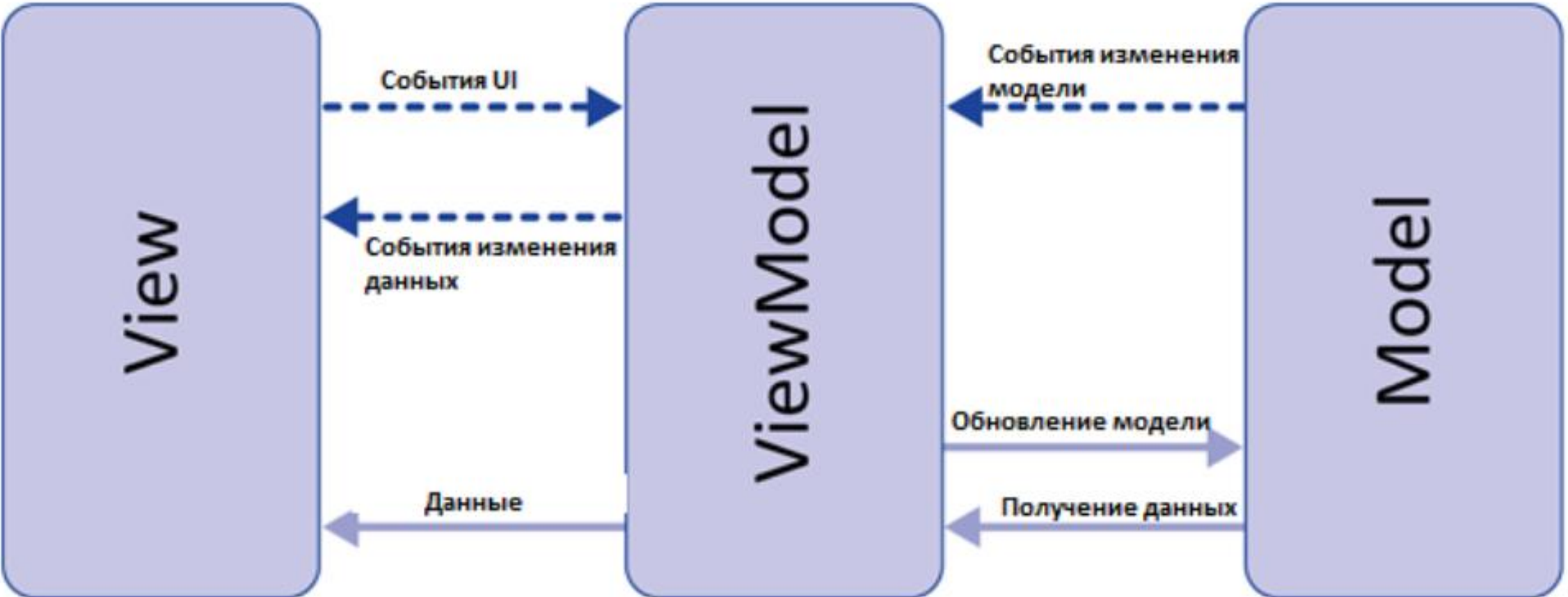
View / Представление

- Отображение данные для пользователя
- Не принимает пользовательский ввод (но это не точно)

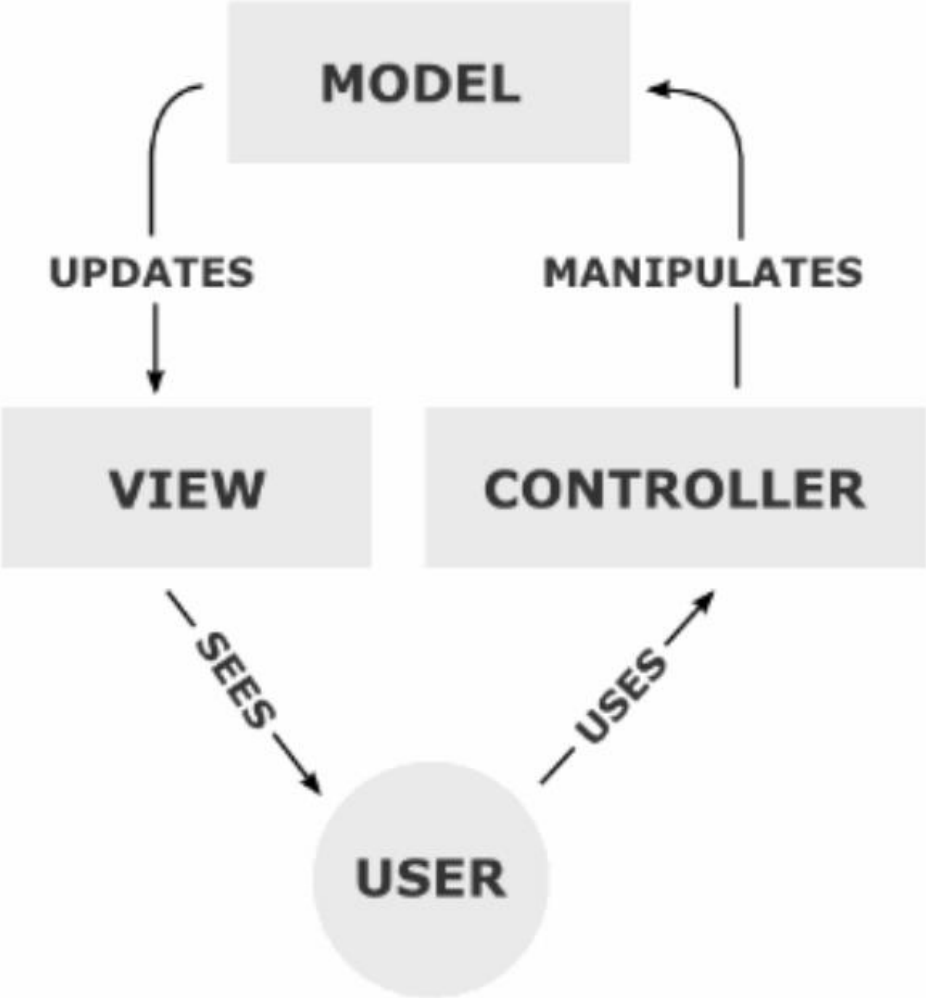
Model-View-Presenter



Model-View-View Model



Model-View-Controller



PROS

- + Абстракции
- + Модульность
- + Скорость разработки
- + Тестируемость
- + Apple одобряет

CONS

- Куда писать код?
- Fat Stupid Ugly Controllers
- Сильное связывание представлений и контроллеров
- Не идеально подходит для современных клиентских приложений

rails - это gem

```
gem install rails
```

```
rails new blog
```


▼ **app**

> **assets**

> **channels**

> **controllers**

> **helpers**

> **javascript**

> **jobs**

> **mailers**

> **models**

> **views**

APP – наше приложение

assets – фронт (css)

channels – веб-сокеты

Controllers

helpers

jobs – задачи по
расписанию

mailers – почта

Models

views

> bin

▼ config

> environments

> initializers

> locales

> webpack

📄 application.rb

📄 boot.rb

📄 YML cable.yml

📄 credentials.yml.enc

📄 YML database.yml

📄 environment.rb

bin – выполняемые скрипты

config

environments – окружения

init. – стартовые скрипты

locales – локализации

webpack – сборка фронта

Окружения

- Development
- Test
- Production
- Stage

▼ public

 404.html

 422.html

 500.html

 apple-touch-icon.png

 apple-touch-icon-precomposed.png

 favicon.ico

 robots.txt

- test
 - channels
 - controllers
 - fixtures
 - helpers
 - integration
 - mailers
 - models
 - system
 - application_system_test_case.rb
 - test_helper.rb

> vendor
.
.browserslistrc
.gitignore
.ruby-version
babel.config.js
config.ru
Gemfile
Gemfile.lock
package.json
postcss.config.js
Rakefile
README.md
yarn.lock

vendor – сторонние
библиотеки

config

environments – окружения

init. – стартовые скрипты

locales – локализации

webpack – сборка фронта

Setup

- Сборка бэк-энда

```
bundle install
```

- Сборка фронта

```
npm install
```

```
yarn install
```

- Подготовка БД

```
bundle exec rake db:create
```

YAML

- *Yet Another Markup Language*
- *YAML Ain't Markup Language*

test:

service: Disk

root: some/folder

Последовательности

- Casablanca
- Spellbound
- Notorious

[milk, bread, eggs, juice]

Сопоставления

Блочный формат

name: John Smith

age: 33

Однострочный формат

{name: John Smith, **age:** 33**}**

Сопоставления из последовательностей

men: [John Smith, Bill Jones]

women:

- Mary Smith
- Susan Williams

Якори & и алиасы *

aliases:

- **&myAlias1**

 - datakey:** dataval 1

 - moredata:** morevals 1

config:

- ***myAlias1**

ССЫЛКИ

https://guides.rubyonrails.org/getting_started.html

<https://habr.com/ru/post/215605/>