

## **МЕТОДИКА РАЗРАБОТКИ АРІ ДЛЯ РАБОТЫ С БАЗОЙ ДАННЫХ ЭЛЕКТРОННОЙ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ СГУГИТ**

***Никита Сергеевич Шлендер***

Сибирский государственный университет геосистем и технологий, 630108, Россия, г. Новосибирск, ул. Плахотного, 10, обучающийся, тел. (960)214-40-19, e-mail: wh3r3areyou@yandex.ru

***Татьяна Юрьевна Бугакова***

Сибирский государственный университет геосистем и технологий, 630108, Россия, г. Новосибирск, ул. Плахотного, 10, заведующая кафедрой прикладной информатики и информационных систем, e-mail: kaf.pi@snga.ru

***Вячеслав Оюнович Сирин-оол***

Сибирский государственный университет геосистем и технологий, 630108, Россия, г. Новосибирск, ул. Плахотного, 10, главный системный администратор, e-mail: info.sgugit@mail.ru

Студентам каждый день необходимо узнавать актуальное расписание занятий и всегда знать о всех изменениях в расписании, и намного удобнее если у них будет сервис, в котором в режиме реального времени будет отображаться вся необходимая им информация и для этой задачи можно реализовать такие инструменты, как мобильное приложение или же frontend приложение, но все эти инструменты не могут существовать без правильно реализованного программного интерфейса приложения или же API. В статье рассматривается анализ видов, функций и протоколов API и этапы разработки API. В ходе разработки был определен основной язык программирования для разработки – PHP и выбраны вспомогательные инструменты для ускоренной разработки – framework Laravel с MVC архитектурой, также выбран инструмент для отображения карты запросов API и описаны методы, используемые в API. Итогом разработки является взаимодействие API с frontend приложением для отображения расписания СГУГИТ.

**Ключевые слова:** API, PHP, Laravel, frontend, электронная информационно-образовательная среда

## **API DEVELOPMENT TECHNIQUE FOR WORKING WITH THE DATABASE OF ELECTRONIC INFORMATION AND EDUCATIONAL ENVIRONMENT SSUGT**

***Nikita S. Schlender***

Siberian State University of Geosystems and Technologies, 10, Plakhotnogo St., Novosibirsk, 630108, Russia, Student, phone: (960)214-40-19, e-mail: wh3r3areyou@yandex.ru

***Tatyana Yu. Bugakova***

Siberian State University of Geosystems and Technologies, 10, Plakhotnogo St., Novosibirsk, 630108, Russia, Ph. D., Head, Department of Applied Computer Science and Information Systems, e-mail: kaf.pi@snga.ru

***Vyacheslav O. Sirin-ool***

Siberian State University of Geosystems and Technologies, 10, Plakhotnogo St., Novosibirsk, 630108, Russia, Chief System Administrator, Department of Applied Computer Science and Information Systems, e-mail: info.sgugit@mail.ru

Every day students need to find out the current class schedule and always be aware of all changes in the schedule, it is much more convenient if they have a service where all the information they need will be displayed in real time and for this task you can implement tools such as a mobile application or the same frontend application, but all these tools cannot exist without a properly implemented application programming interface or API. The article discusses the analysis of API types, functions and protocols and the stages of API development. In the course of development, the main programming language for development, PHP, was chosen and auxiliary tools for accelerated development were selected - the Laravel framework with MVC architecture, and a tool for displaying a map of API requests was also selected; the methods used in the API were described. The result of the development is the interaction of the API with the frontend application for displaying the SSUGT schedule.

**Keywords:** API, PHP, Laravel, frontend, electronic information and educational environment

### *Введение*

Реализация API происходит на стороне backend приложения и для этого необходимо выбрать вид API, необходимо определить методы, которые будут доступны в API и по какому протоколу будет взаимодействие сторонних сервисов с API, а также определить основной язык программирования и вспомогательные инструменты для ускорения разработки.

API (Application Programming Interface) – это интерфейс, позволяющий двум независимым компонентам программного обеспечения обмениваться информацией. API играет роль посредника между внутренними и внешними программными функциями, обеспечивая настолько эффективный обмен информацией, что конечные пользователи обычно его просто не замечают [1].

API по своей сути это набор методов, которые выполняются по определенному запросу от внешнего пользователя и выполняют заданные им действия, и возвращают ответ пользователю, который выполнял запрос к API.

API бывают двух видов: частные и открытые API.

Частные API недоступны разработчикам, которые не имеют отношения к разработке определенного продукта или же не относятся к сотрудникам компании. Такие API необходимы для взаимодействия в работе только сотрудников компании.

Открытые API, в свою очередь, предоставляют доступ внешним разработчикам к получению определенной информации по API, к интеграции данных внешних разработчиков с продуктом.

Открытые или частные API экономят время разработчиков, позволяя им объединять платформы с имеющимися инструментами и устраняя необходимость в создании нового функционала с нуля.

Протоколы, через которые реализуется API – SOAP и REST.

SOAP (Simple Object Access Protocol) до недавнего времени считался безусловным фаворитом у разработчиков API. Но сейчас 70 % открытых API соответствуют протоколу REST. SOAP по-прежнему используется во многих круп-

ных технических компаниях и обеспечивает поддержку устаревших систем, которые могут быть совместимы только с ним [2].

REST (Representational State Transfer) – это новый протокол веб-сервисов, позволяющий работать с большим количеством форматов данных. Кроме того, REST предпочтительнее для разработчиков, так как предлагает меньшее время загрузки и более высокую эффективность.

В отличие от SOAP, который работает с форматом XML, RESTful поддерживает JSON, XML, YAML и тд.

### ***Язык программирования***

Разработка API была выполнена на языке программирования PHP, так как существует множество готовых инструментов на этом языке программирования, которые реализуют MVC-шаблон программирования, с помощью которого можно реализовать правильную архитектуру реализации CRUD-операций над данными по определенным запросам API.

### ***Вспомогательные инструменты для разработки***

Для ускорения разработки необходим framework, который бы поддерживал систему роутингов и представлял MVC архитектуру для удобной работы с CRUD операциями.

Фреймворк – это программный продукт, который упрощает создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надежность решений [3].

Для разработки был выбран популярный бесплатный framework – Laravel.

Laravel – это бесплатный PHP фреймворк общего назначения с MVC шаблоном проектирования, который появился на свет сравнительно недавно – в 2011 году, но, благодаря стремительным темпам развития и огромной армии поклонников, сегодня он является одним из самых популярных PHP движков [4].

Вспомогательным инструментом для создания карты запросов API в данном проекте является Postman.

Postman – основное предназначение приложения – создание коллекций с запросами к вашему API. Любой разработчик или тестировщик, открыв коллекцию, сможет с лёгкостью разобраться в работе вашего сервиса [5].

### ***Результаты***

Пример одного из запросов API описанного в Postman представлен на рис. 1–3.

POST /api/v1/schedule/get-schedule

/api/v1/schedule/get-schedule

Получение расписания за определенную неделю и для определенной группы

BODY formdata

id_week	192
id_SubGroup	492

The screenshot shows a REST client interface with the following details:

- Endpoint:** /api/v1/schedule/get-schedule
- Method:** POST
- Request Body:** formdata
- Request Parameters:** id\_week: 192, id\_SubGroup: 492
- Example Request Code:**

```
<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('/api/v1/schedule/get-schedule');
$request->setMethod(HTTP_Request2::METHOD_POST);
$request->setConfig(array(
    'follow_redirects' => TRUE
));
$request->addPostParameter(array(
    'id_week' => '192',
```
- Example Response:** 200 OK
- Response Body:** A JSON array with one element:

```
{
  "1": {
    "1": {
      "subGroup": "ЭН-11.1",
      "group": "ЭН-11",
      "cabinet": "425",
      "teacher": "Жданов Сергей Сергеевич",
      "discipline": "3D-прототипирование",
      "numberLesson": {
        "id_Para": "65",
        "ParaNomer": "1",
        "BegPara": "08:30:00.0000000",
        "EndPara": "10:00:00.0000000",
        "id_God": "13",
        "Sem": "1"
      }
    }
  }
}
```

Рис. 1. Пример одного из методов API

The screenshot shows the full JSON response from the API:

```
{
  "1": {
    "1": {
      "subGroup": "ЭН-11.1",
      "group": "ЭН-11",
      "cabinet": "425",
      "teacher": "Жданов Сергей Сергеевич",
      "discipline": "3D-прототипирование",
      "numberLesson": {
        "id_Para": "65",
        "ParaNomer": "1",
        "BegPara": "08:30:00.0000000",
        "EndPara": "10:00:00.0000000",
        "id_God": "13",
        "Sem": "1"
      }
    },
    "week": {
      "id_Nedel": "192",
      "NomerNedelSem": "36",
      "NomerNedelGod": "36",
      "Chet": "1",
      "Sem": "1",
      "id_God": "13",
      "DateNedel": "2021-05-03 00:00:00",
      "NomerNedelRasp": "1"
    },
    "day": {
      "id_Nedel": "192",
      "NomerNedelSem": "36",
      "NomerNedelGod": "36",
      "Chet": "1",
      "Sem": "1",
      "id_God": "13",
      "DateNedel": "2021-05-03 00:00:00",
      "NomerNedelRasp": "1"
    }
  },
  "2": {
    "subGroup": "ЭН-11.1",
    "group": "ЭН-11",
    "cabinet": "423",
    "teacher": "Максименко Любовь Александровна",
    "discipline": "Автоматизированные системы технической подготовки производства",
    "numberLesson": {
```

Рис. 2. Пример результата запроса одного из методов API

ЭН-11.1, 1 неделя

03.05 - 09.05

08:30 10:00	3D-прототипирование 425 Жданов Сергей Сергеевич лекция		
10:10 11:40	Автоматизированные системы технической подготовки производства 423 Максименко Любовь Александровна лекция		
12:10 13:40	Геодезические методы изучения геодинамических процессов 432 Луговская Анна Юрьевна лекция		
13:50 15:20	Биология почв 301 Чернышева Ольга Владимировна лекция		06.05
03.05			

Рис. 3. Пример отображения результата API на frontend приложении

### *Заключение*

В результате было реализовано backend приложение, которое принимает запросы по определенному маршруту, обрабатывает запросы и выполняет нужный метод для взаимодействий с базой данных электронной информационно-образовательной средой СГУГиТ, возвращая ответ клиенту. Имея полную карту запросов и описание работы методов API в будущем можно создать новое frontend приложение для взаимодействия с API.

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Что такое программный интерфейс приложений (API) [Электронный ресурс] – Режим доступа: <https://www.wrike.com/ru/blog/chto-takoe-programmnyj-interfejs-prilozhenij-api>.
2. Рельсы веб-интеграции. REST и SOAP. [Электронный ресурс] – Режим доступа: <https://www.intervolga.ru/blog/projects/relsy-veb-integratsii-rest-i-soap>.
3. Фреймворки в веб-разработке. [Электронный ресурс] – Режим доступа: [https://web-creator.ru/articles/about\\_frameworks](https://web-creator.ru/articles/about_frameworks).
4. Laravel – это что? [Электронный ресурс] – Режим доступа: <http://web.spt42.ru/index.php/chto-takoe-laravel>.
5. Введение в Postman. [Электронный ресурс] – Режим доступа: <https://www.pvsm.ru/api/275327>.

© Н. С. Шлендер, Т. Ю. Бугакова, В. О. Сирин-оол, 2021