

Реализация API

Теоретическая часть

- Формат JSON;
- Стандарт JSON-RPC;
- Декораторы в Python;
- Использование JSON-RPC в Flask;
- Валидация параметров запросов, формы;
- Обработка ошибок в Web-проектах;

Практическая часть

- Работа с JSON в Python;
- Проектирование API проекта;
- Тестирование API проекта.

Application programming interface (API)

Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

Виды совместимости приложений

- обратная совместимость — более новый код способен читать данные, записанные более старым;
- прямая совместимость — более старый код способен читать данные, записанные более новым.

Форматы передачи данных

- Текстовые форматы (JSON, XML, CSV);
- Бинарный формат (Apache Thrift, Protocol Buffers);

Формат JSON

Формат JSON

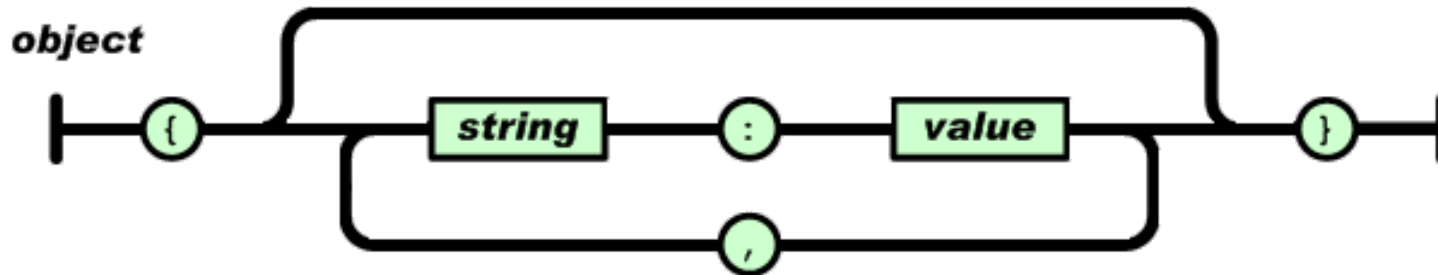
JSON (англ. JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript.

```
{  
  "firstName": "Иван",  
  "lastName": "Иванов",  
  "phoneNumbers": [  
    "812 123-1234",  
    "916 123-4567"  
  ]  
}
```


Формат JSON

JSON основан на двух структурах данных:

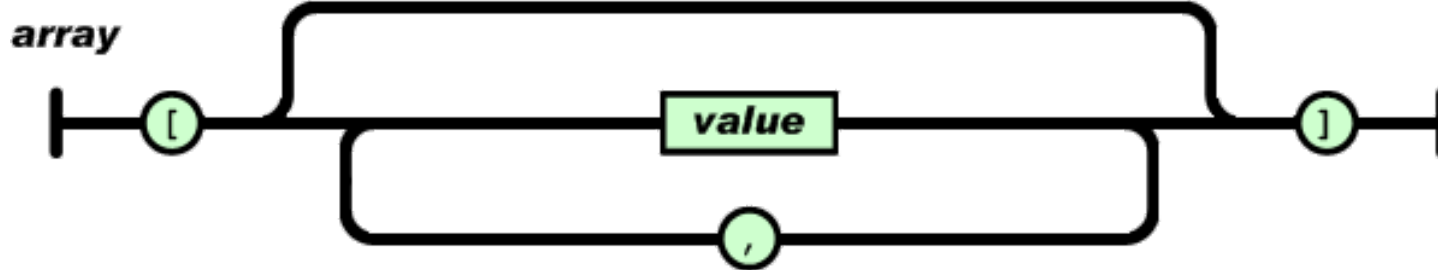
- Коллекция пар ключ/значение. В разных языках, эта концепция реализована как объект, запись, структура, словарь, хэш, именованный список или ассоциативный массив;



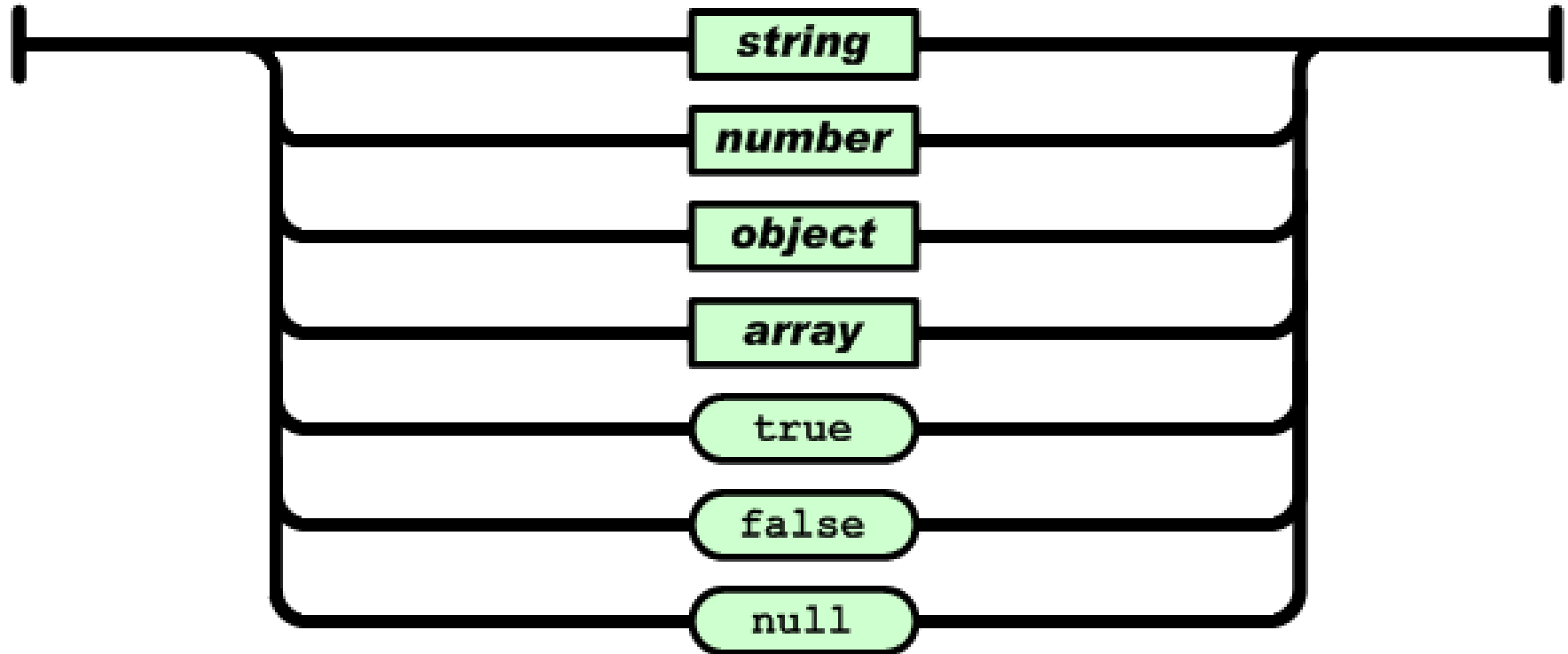
Формат JSON

JSON основан на двух структурах данных:

- Упорядоченный список значений. В большинстве языков это реализовано как массив, вектор, список или последовательность.



value



Формат JSON

Преимущества:

- Легко читается человеком;
- Компактный;
- Для работы с JSON есть множество библиотек;
- Больше структурной информации в документе.

Формат JSON

- JSON - это формат данных - он содержит только свойства, а не методы;
- JSON требует двойных кавычек, которые будут использоваться вокруг строк и имен свойств;
- Вы можете проверить JSON с помощью приложения, такого как jsonlint;
- JSON может фактически принимать форму любого типа данных, который действителен для включения внутри JSON, а не только массивов или объектов.

Стандарт JSON- RPC

Стандарт JSON-RPC

JSON-RPC (сокр. от англ. JavaScript Object Notation Remote Procedure Call — JSON-вызов удалённых процедур) — протокол удалённого вызова процедур, использующий JSON для кодирования сообщений.

Стандарт JSON-RPC

Формат входного запроса:

- `method` — строка с именем вызываемого метода;
- `params` — массив объектов, которые должны быть переданы методу, как параметры;
- `id` — значение любого типа, которое используется для установки соответствия между запросом и ответом.

Стандарт JSON-RPC

Формат ответа:

- `result` — данные, которые вернул метод. Если произошла ошибка во время выполнения метода, это свойство должно быть установлено в `null`;
- `error` — код ошибки, если произошла ошибка во время выполнения метода, иначе `null`;
- `id` — то же значение, что и в запросе, к которому относится данный ответ.

Стандарт JSON-RPC

Пример запроса:

```
{ "method": "echo", "params": ["Hello JSON-RPC"], "id": 1 }
```

Пример ответа:

```
{ "result": "Hello JSON-RPC", "error": null, "id": 1 }
```

Декораторы в Python

Декораторы в Python

Это функция, которая принимает функцию в качестве единственного аргумента и возвращает новую функцию, с дополнительными функциональными возможностями.

```
def my_decorator(function):  
    def wrapper(*args, **kwargs):  
        return result  
    return wrapper
```

```
@my_decorator
```

```
def foo():
```

```
    # тело функции
```

Параметризуемый декораторы в Python

Вызываем функцию с требуемыми параметрами, и она вернёт декоратор, который будет использован для декорирования следующей за ним функцией.

Домашнее задание

- Переработать API с использованием Flask-JSONRPC (3 балла);
- Реализовать методы и тесты для (4 балла):
 - отправка сообщения;
 - получение списка сообщений чата;
 - прочтение сообщения.
- Валидировать входные параметры API (3 балла).

Срок сдачи: следующее занятие.

Спасибо за внимание!

