

Форматы обмена данными

Интерфейс (от англ. *interface*) — это посредник, средство взаимодействия двух систем. Например, регулятор громкости радиоприёмника — это интерфейс регулирования громкости, посредник между системами «человек» и «электронная начинка приёмника».

Графическая оболочка операционной системы компьютера — это тоже интерфейс. На экране нарисованы кнопки, пользователь нажимает на них, решая свои повседневные задачи, и при этом не задумывается о внутреннем устройстве самой операционной системы.

При перетаскивании иконки файла на иконку «корзина» файл удаляется, в файловой системе происходят изменения, но сама процедура перетаскивания картинок на экране — это лишь удобный посредник, «кнопка для запуска» скрытых от пользователя процессов в системе.

API (от англ. Application Programming Interface, «программный интерфейс приложения») — это интерфейс для обмена данными. Слово «программный» означает, что API служат в первую очередь для взаимодействия программ: с системой взаимодействует не разработчик, а код, написанный им.

Вы уже работали с API в вводном курсе: Анфиса отправляла запрос к API и получала информацию о погоде на завтра.

API могут общаться друг с другом: например, ваш код запрашивает данные с API Яндекса Маркета о появлении новых товаров, а Яндекс Маркет, в свою очередь, обращается к API различных магазинов, чтобы эти данные получить.

Программу, которая обращается к API с запросом, называют «клиент» (или *client*, если по-английски). Клиентом может быть код на сервере, мобильное приложение, программа для тестирования или даже обычный веб-браузер.

При этом каждый сервис может быть написан на своём языке программирования, но благодаря API сервисы могут легко общаться между собой по сети, используя протокол HTTP и передавая данные в удобном для всех формате.



JSON

Один из самых распространённых форматов передачи данных — это JSON.

JSON расширяется как *JavaScript Object Notation* (англ. «объектная запись JavaScript»). Исторически сложилось так, что этот формат «вырос» из языка программирования *JavaScript*. По структуре JSON очень похож на тип данных `dict`: это последовательность пар «ключ-значение»; как и словари, JSON поддерживает вложенность. Но JSON более стандартизирован: например, ключи словаря в JSON пишутся только в двойных кавычках. Значениями ключей могут быть строки, числа, булевы значения, словари и списки.

Например, досье на супергероя в формате JSON может выглядеть так:

```
JSON
{
  {
    "name": "Captain America",
    "realName": "Steve Rogers",
    "yearCreated": 1941,
    "powers": [
      "Strength",
      "Healing ability"
    ]
  }
}
```

А досье на нескольких супергероев в формате JSON может уже выглядеть так:

```
JSON
[
  {
    {
      "name": "Captain America",
      "realName": "Steve Rogers",
      "yearCreated": 1941,
      "powers": [
        "Strength",
        "Healing ability"
      ]
    },
    {
      "name": "Spider-Man",
      "realName": "Peter Parker",
      "yearCreated": 1963,
      "powers": [
        "Danger sense",
        "Speed",
        "Jumping"
      ]
    }
  ]
}
```

Некоторые типы данных в JSON не формализованы: например, в нём нет специального формата для даты и данные можно передавать в любом удобном формате — например, как строку `"09-10-1988"` или как целое число `1623799668`.



Более детально о формате JSON можно узнать [из документации](#).



Иногда бывает трудно на глаз определить, где именно спряталась ошибка, — ведь в JSON объекты могут быть вложены один в другой. Чтобы убедиться в том, что JSON правильно составлен, можно обратиться к любому онлайн-валидатору, например [вот к этому](#).

XML

JSON — популярный, но не единственный формат для передачи данных.

XML (*eXtensible Markup Language*, «расширяемый язык разметки») тоже весьма популярен и широко применяется в разработке. Внешне этот язык чем-то похож на HTML, и это не случайно: язык разметки веб-страниц — прямой потомок XML.

Одно из основных отличий HTML от XML в том, что названия тегов в XML не стандартизированы, их можно называть по собственному желанию. Да и предназначение у этих языков разное: теги HTML служат для отображения информации в браузере, а XML-теги просто структурируют передаваемую информацию.

Синтаксис похож на HTML: теги пишутся в треугольных скобках, есть открывающий тег, и (в большинстве случаев) к нему в пару должен быть закрывающий.

Вот та же картотека супергероев, но в формате XML:

```
XML
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <superhero>
    <name>Captain America</name>
    <realName>Steve Rogers</realName>
    <yearCreated>1941</yearCreated>
    <powers>Strength</powers>
    <powers>Healing ability</powers>
  </superhero>
  <superhero>
    <name>Spider-Man</name>
    <realName>Peter Parker</realName>
    <yearCreated>1963</yearCreated>
    <powers>Danger sense</powers>
    <powers>Speed</powers>
    <powers>Jumping</powers>
  </superhero>
</root>
```

Проверить валидность XML можно с помощью любого онлайн-валидатора, например через валидатор на [codebeautify.org](#).



Более подробно об XML можно почитать [в документации](#).



С чем будем работать?

Для обмена данными в API может применяться JSON, XML, YAML или любой другой формат, поддерживающий структурирование данных. Зачастую разработчики включают в свои API поддержку нескольких форматов обмена данными.

JSON — один из наиболее популярных форматов обмена данными при работе с API. Он проще для чтения и при равном объёме данных весит чуть меньше, чем XML.

JSON поддерживают все популярные языки программирования; в этом курсе вы будете работать в основном с ним.

Что из перечисленного можно назвать интерфейсом?

- ☐ Розетка
- Правильно, розетка — интерфейс для взаимодействия с системой энергоснабжения.
- ☒ Клавиатура компьютера
- Правильно: нажимая на клавиши, мы взаимодействуем с компьютером.
- ☒ Выключатель света в ванной
- Правильно, через выключатель мы управляем системой освещения в ванной.
- ☐ Утюг
- ☐ Горшок с геранью

2 из 5 правильно и 0 неправильно

Провалидируйте XML, всё ли с ним в порядке? (Воспользуйтесь любым онлайн-валидатором.)

```
XML
<?xml version="1.0" encoding="UTF-8"?>
<message>
  <to>You</to>
  <from>Me</from>
  <text>Привет!</text>
</message>
```

- ☐ Да, это валидный XML
- ☒ В коде ошибка, тут чего-то не хватает
- Правильно, нужно закрыть тег `<message>` в самом конце.

1 из 2 правильно и 0 неправильно

Провалидируйте JSON, всё ли с ним в порядке? (Воспользуйтесь любым онлайн-валидатором.)

```
JSON
{
  "name": "Captain America",
  "realName": "Steve Rogers",
  "yearCreated": 1941,
  "powers": [
    "Strength",
    "Healing ability"
  ]
}
```

- ☒ Да, это валидный JSON
- Нет, тут ошибка: кавычки должны быть двойные.
- ☐ Нет, тут что-то не так
- Правильно, одиночные кавычки нельзя использовать.

0 из 2 правильно и 1 неправильно