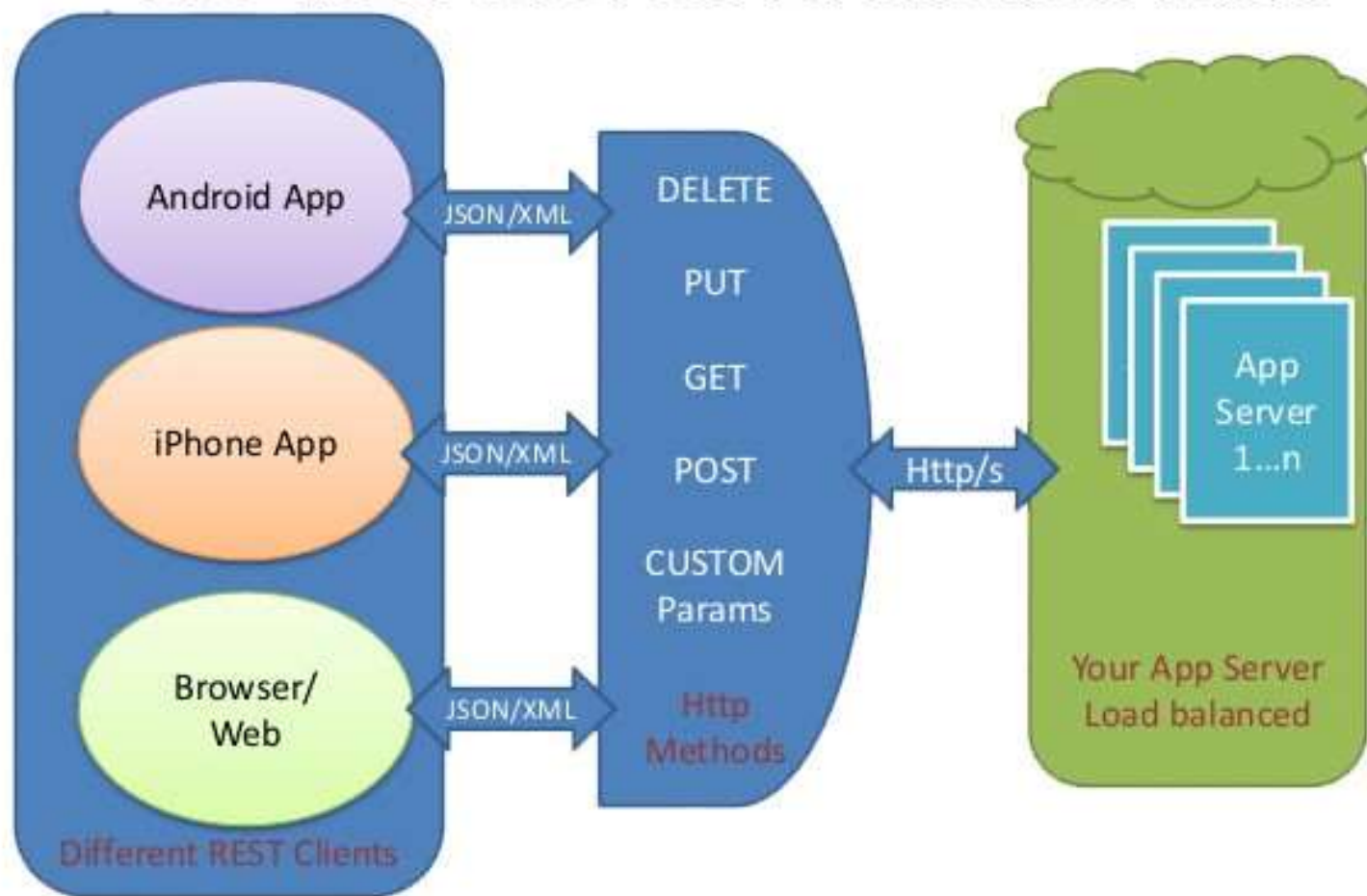


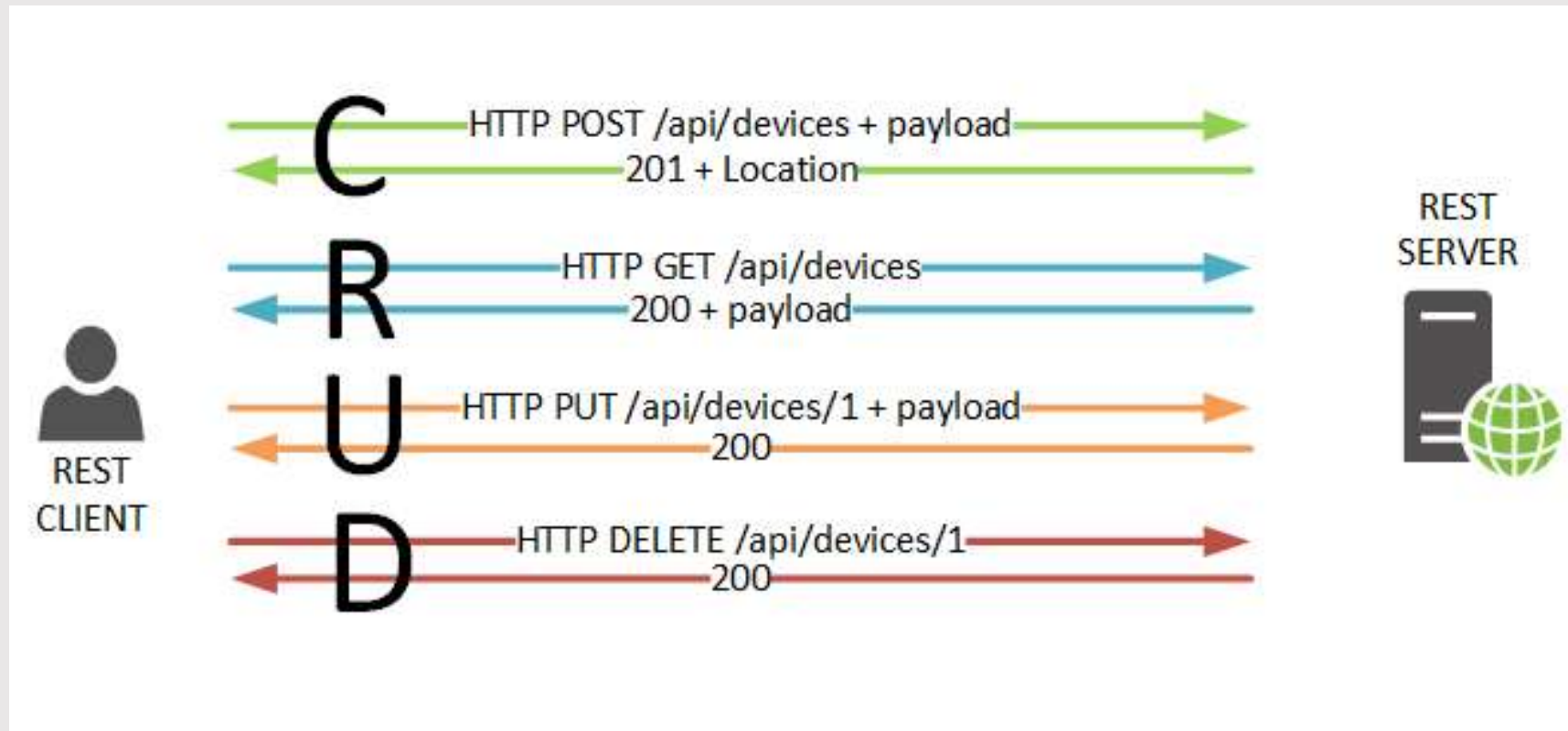
REST API Architecture



Команды манипулирования ресурсом

Представление	Описание
GET	Извлекает представление ресурса
HEAD	Идентично GET, но без тела ответа; обычно используется для получения заголовка
POST	Создает новый ресурс
PUT	Обновляет ресурс
DELETE	Удаляет ресурс
OPTIONS	Извлекает разрешенные HTTP-методы

REST API



JSON формат

```
> [  
>   {  
>     "id": 1,  
>     "title": "Новость",  
>     "category": "Политика",  
>     "description": " Описание новости ",  
>     "source": "http://hotnews.com/12",  
>     "datetime": 1611579303  
>   },  
>   {  
>     "id": 2,  
>     "title": «Другая новость»,  
>     "category": «Спорт»,  
>     "description": «Победы наших спортсменов»,  
>     "source": "http://sport.ru",  
>     "datetime": 1610993312  
>   }  
> ]
```

Реализация REST сервиса (PHP)

.htaccess (маршрутизация)

RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f

RewriteRule ^(.+)\$ index.php?path=\$1 [L,QSA]

Получение данных запроса

```
if ($_SERVER['REQUEST_METHOD'] === 'GET') return $_GET;  
else return file_get_contents('php://input');
```

Формирование ответа

```
header("Content-Type: application/json; charset=UTF-8");
```

Функции форматирования

```
json_encode(объект или массив)
```

```
json_decode( $json_string, true /* объект или массив*/);
```

Реализация клиента (PHP)

Необходимо выполнить HTTP запрос с нужными параметрами:

```
$sURL = "http://mysite.com/api/news"; // URL-адрес POST
$sPD = json_encode($myObj); // Данные POST
$aHTTP = array(
    'http' => // Обертка, которая будет использоваться
        array(
            'method' => 'POST', // Метод запроса
            // Ниже задаются заголовки запроса
            'header' => 'Content-type: application/x-www-form-urlencoded',
            'content' => $sPD
        )
    );
$context = stream_context_create($aHTTP);
$contents = file_get_contents($sURL, false, $context);
echo $contents;
```

Fetch API (JavaScript)

Функция **fetch()** позволяет выполнить запрос на сервер

- Для запросов методом GET параметры указываются в адресной строке
- Для запросов методом POST, PUT, PATCH данные передаются в теле запроса (body) в выбранном формате (json).

```
fetch(url, {  
  body: JSON.stringify(data),  
  credentials: "include",  
  headers: {  
    "Content-Type": "application/json"  
  },  
  method: "POST"  
});
```

Fetch API (продолжение)

- При вызове функции **fetch()**, она возвращает **Promise** объект.
- У этого объекта (**Promise**) можно вызвать метод `then()` получив в callback функции объект **Response**
- Объект **Response** позволить получить данные с сервера, например вызовом метода **response.json()**.

```
fetch("/rest/news")  
  .then(response => response.json())  
  .then(data => {  
    console.log(data);  
  })  
  .catch(error => alert(error) );
```