

Web Development III

Block: V

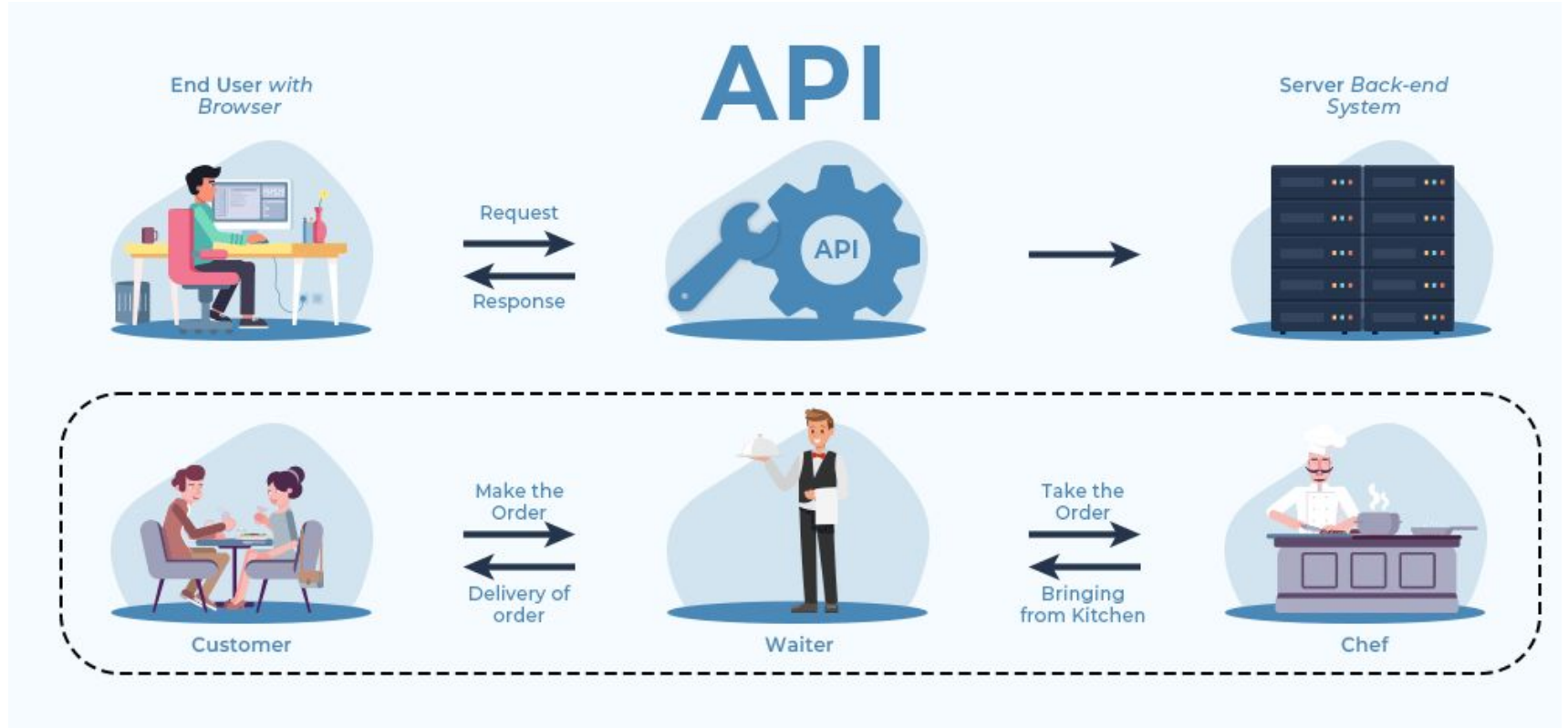


Block: V - Agenda

- Introduction to API
- Types of API
- How API works
- Simple RESTful API with PHP and MySQL



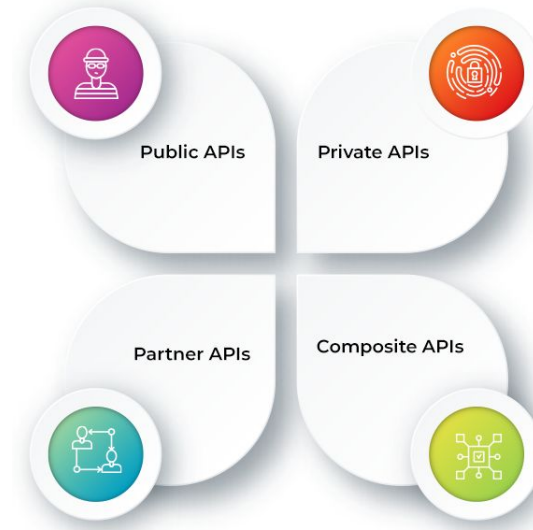
Introduction to API



Introduction to API

- API stands for Application Programming Interface, is a set of defined rules that enable different applications to communicate with each other.
- An API defines how a developer should request services from an operating system (OS) or other application, and expose data within different contexts and across multiple channels.
- APIs let your product or service communicate with other products and services without having to know how they're implemented.
- This can simplify app development, saving time and money. When you're designing new tools and products—or managing existing ones—APIs give you flexibility; simplify design, administration, and use; and provide opportunities for innovation.
- Types of Web APIs: Web APIs are those which are accessible over the internet.
 - Open APIs: These APIs are publicly available as there is no restrictions.
 - Partner APIs: The user needs license and special rights to access this type of APIs.
 - Private APIs: Owned by companies for internal systems.
 - Composite APIs: It's a combination of data and service APIs for speeding up the execution process.

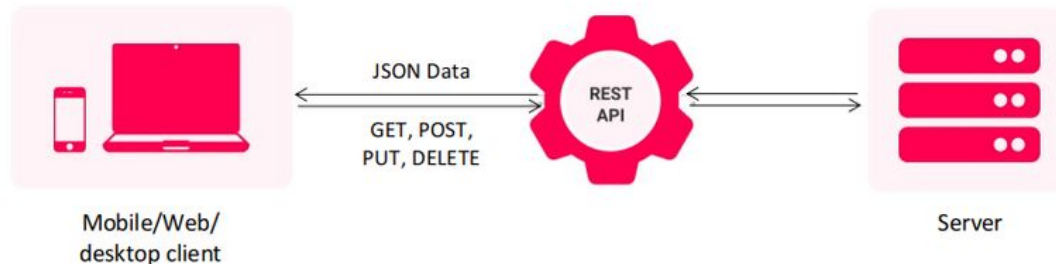
TYPES OF APIS



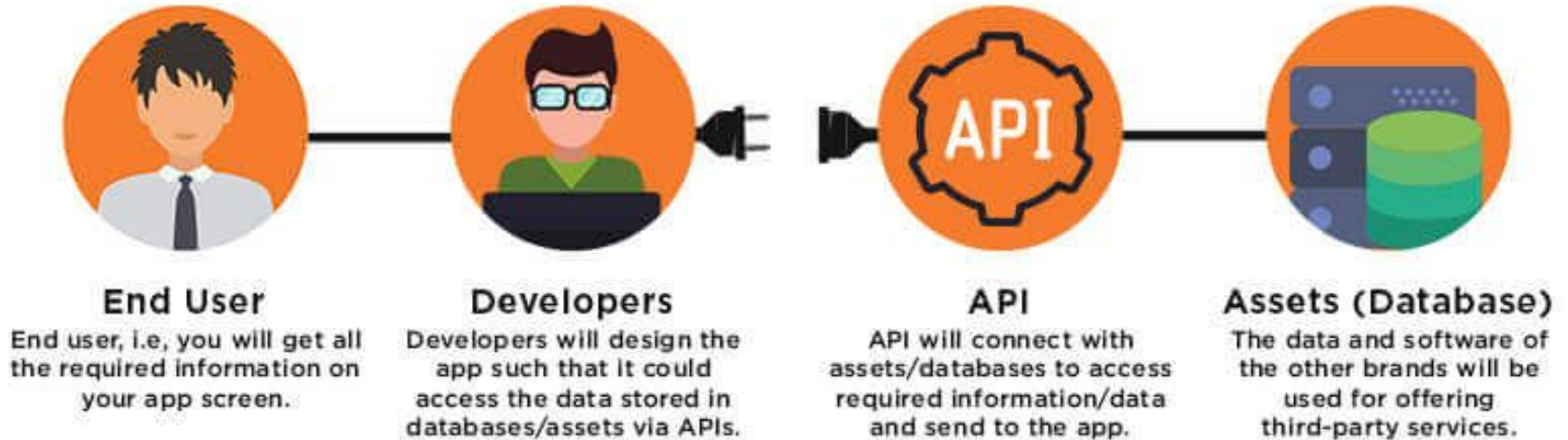
Types of API

- SOAP: Simple Object Access Protocol which uses web service definition language or XML for data transfer. It is very robust. These are used in integrating APIs.
 - JSON: For data transfer it uses JSON.
 - REST: The set of rules includes some standard architectural principles for data exchange. For making a request, it uses HTTP methods of getting, PUT, POST, GET, DELETE for all CRUD operations. It consumes less bandwidths and also comfortable accessing cloud services.
-
- REST output is in the form of JSON:
 - GET: Read or retrieves information.
 - POST: Creates new record.
 - PUT: Update a record.
 - DELETE: Deletes a record.

REST API Model

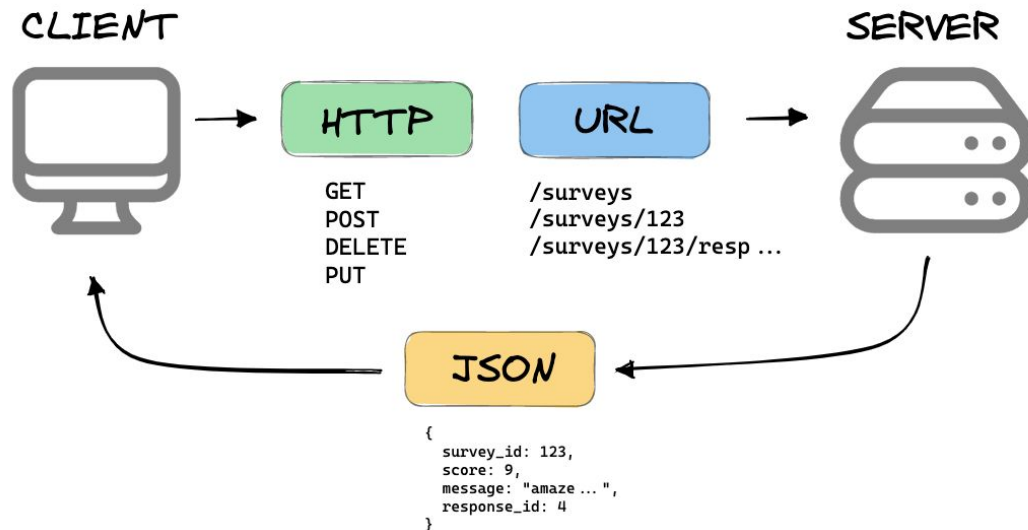
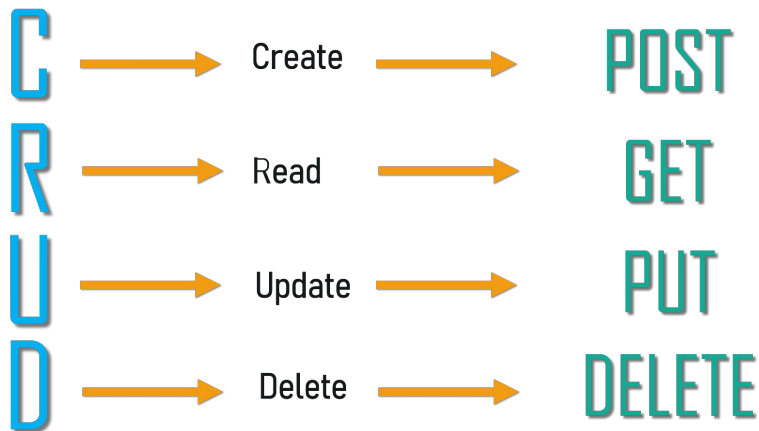


Working of API



Simple RESTful API with PHP and MySQL

- Create database table
- Connect database
- Create a file for REST API
 - Create
 - Read
 - Update
 - Delete
- Check Endpoint in REST-Client (POSTMAN)



REST-API: Create database table

```
CREATE TABLE `items` (  
  `id` int(11) NOT NULL,  
  `name` varchar(256) NOT NULL,  
  `description` text NOT NULL,  
  `price` int(255) NOT NULL,  
  `category_id` int(11) NOT NULL,  
  `created` datetime NOT NULL,  
  `modified` timestamp NOT NULL DEFAULT  
    CURRENT_TIMESTAMP  
) ENGINE=MyISAM DEFAULT  
CHARSET=utf8;
```

The screenshot displays the phpMyAdmin interface for a database named 'api_demo'. The left sidebar shows the database structure, with 'items' selected under the 'api_demo' database. The main panel shows the 'items' table structure and a list of 16 rows of data. The table has columns: id, name, description, price, category_id, created, and modified. The data includes various products like LG P880 4X HD, Google Nexus 4, Samsung Galaxy S4, Bench Shirt, Lenovo Laptop, Samsung Galaxy Tab 10.1, Spalding Watch, Sony Smart Watch, Huawei Y300, Abercrombie Lake Arnold Shirt, Abercrombie Allen Brook Shirt, Another product, Wallet, Amanda Waller Shirt, Nike Shoes for Men, Bristol Shoes, and Rolex Watch.

	id	name	description	price	category_id	created	modified
<input type="checkbox"/>	1	LG P880 4X HD	My first awesome phone!	336	3	2022-06-01 01:12:26	2023-05-31 17:42:26
<input type="checkbox"/>	2	Google Nexus 4	The most awesome phone of 2013!	299	2	2022-06-01 01:12:26	2023-05-31 17:42:26
<input type="checkbox"/>	3	Samsung Galaxy S4	How about no?	600	3	2022-06-01 01:12:26	2023-05-31 17:42:26
<input type="checkbox"/>	6	Bench Shirt	The best shirt!	29	1	2022-06-01 01:12:26	2023-05-31 02:42:21
<input type="checkbox"/>	7	Lenovo Laptop	My business partner.	399	2	2022-06-01 01:13:45	2023-05-31 02:43:39
<input type="checkbox"/>	8	Samsung Galaxy Tab 10.1	Good tablet.	259	2	2022-06-01 01:14:13	2023-05-31 02:44:08
<input type="checkbox"/>	9	Spalding Watch	My sports watch.	199	1	2022-06-01 01:18:36	2023-05-31 02:48:31
<input type="checkbox"/>	10	Sony Smart Watch	The coolest smart watch!	300	2	2022-06-06 17:10:01	2023-06-05 18:39:51
<input type="checkbox"/>	11	Huawei Y300	For testing purposes.	100	2	2022-06-06 17:11:04	2023-06-05 18:40:54
<input type="checkbox"/>	12	Abercrombie Lake Arnold Shirt	Perfect as gift!	60	1	2022-06-06 17:12:21	2023-06-05 18:42:11
<input type="checkbox"/>	13	Abercrombie Allen Brook Shirt	Cool red shirt!	70	1	2022-06-06 17:12:59	2023-06-05 18:42:49
<input type="checkbox"/>	26	Another product	Awesome product!	555	2	2022-11-22 19:07:34	0000-00-00 00:00:00
<input type="checkbox"/>	28	Wallet	You can absolutely use this one!	799	6	2022-12-04 21:12:03	2023-12-03 23:42:03
<input type="checkbox"/>	31	Amanda Waller Shirt	New awesome shirt!	333	1	2022-12-13 00:52:54	2023-12-12 03:22:54
<input type="checkbox"/>	42	Nike Shoes for Men	Nike Shoes	12999	3	2022-12-12 06:47:08	2023-12-12 07:17:08
<input type="checkbox"/>	48	Bristol Shoes	Awsome shoes.	999	5	2022-01-08 06:36:37	2023-01-08 07:06:37
<input type="checkbox"/>	60	Rolex Watch	Luxury watch.	25000	1	2022-01-11 15:46:02	2023-01-11 16:16:02

REST-API: Create database table

```
INSERT INTO `items` (`id`, `name`,  
`description`, `price`, `category_id`,  
`created`, `modified`) VALUES  
(1, 'LG P880 4X HD', 'My first awesome phone!',  
336, 3, '2022-06-01 01:12:26', '2023-05-31  
17:42:26'),  
(2, 'Google Nexus 4', 'The most awesome phone  
of 2013!', 299, 2, '2022-06-01 01:12:26',  
'2023-05-31 17:42:26'),....)
```

```
ALTER TABLE `items`  
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `items`  
MODIFY `id` int(11) NOT NULL  
AUTO_INCREMENT, AUTO_INCREMENT=61;
```

The screenshot displays the phpMyAdmin interface. On the left is the database navigation tree with 'api_demo' selected. The main panel shows the 'Table: items' view. A green status bar indicates 'Showing rows 0 - 16 (17 total, Query took 0.0002 seconds.)'. Below this, a SQL query 'SELECT * FROM `items`' is shown. A toolbar contains options like 'Show all', 'Number of rows: 25', 'Filter rows: Search this table', and 'Sort by key: None'. The 'Extra options' section is expanded. The table data is displayed in a grid with columns: id, name, description, price, category_id, created, and modified. Each row includes action icons for Edit, Copy, and Delete.

	id	name	description	price	category_id	created	modified
<input type="checkbox"/>	1	LG P880 4X HD	My first awesome phone!	336	3	2022-06-01 01:12:26	2023-05-31 17:42:26
<input type="checkbox"/>	2	Google Nexus 4	The most awesome phone of 2013!	299	2	2022-06-01 01:12:26	2023-05-31 17:42:26
<input type="checkbox"/>	3	Samsung Galaxy S4	How about no?	600	3	2022-06-01 01:12:26	2023-05-31 17:42:26
<input type="checkbox"/>	6	Bench Shirt	The best shirt!	29	1	2022-06-01 01:12:26	2023-05-31 02:42:21
<input type="checkbox"/>	7	Lenovo Laptop	My business partner.	399	2	2022-06-01 01:13:45	2023-05-31 02:43:39
<input type="checkbox"/>	8	Samsung Galaxy Tab 10.1	Good tablet.	259	2	2022-06-01 01:14:13	2023-05-31 02:44:08
<input type="checkbox"/>	9	Spalding Watch	My sports watch.	199	1	2022-06-01 01:18:36	2023-05-31 02:48:31
<input type="checkbox"/>	10	Sony Smart Watch	The coolest smart watch!	300	2	2022-06-06 17:10:01	2023-06-05 18:39:51
<input type="checkbox"/>	11	Huawei Y300	For testing purposes.	100	2	2022-06-06 17:11:04	2023-06-05 18:40:54
<input type="checkbox"/>	12	Abercrombie Lake Arnold Shirt	Perfect as gift!	60	1	2022-06-06 17:12:21	2023-06-05 18:42:11
<input type="checkbox"/>	13	Abercrombie Allen Brook Shirt	Cool red shirt!	70	1	2022-06-06 17:12:59	2023-06-05 18:42:49
<input type="checkbox"/>	26	Another product	Awesome product!	555	2	2022-11-22 19:07:34	0000-00-00 00:00:00
<input type="checkbox"/>	28	Wallet	You can absolutely use this one!	799	6	2022-12-04 21:12:03	2023-12-03 23:42:03
<input type="checkbox"/>	31	Amanda Waller Shirt	New awesome shirt!	333	1	2022-12-13 00:52:54	2023-12-03 02:22:54
<input type="checkbox"/>	42	Nike Shoes for Men	Nike Shoes	12999	3	2022-12-12 06:47:08	2023-12-12 07:17:08
<input type="checkbox"/>	48	Bristol Shoes	Awesome shoes.	999	5	2022-01-08 06:36:37	2023-01-08 07:06:37
<input type="checkbox"/>	60	Rolex Watch	Luxury watch.	25000	1	2022-01-11 15:46:02	2023-01-11 16:16:02

REST-API: Connect to database

Database.php

```
<?php
class Database{

    private $host = 'localhost';
    private $user = 'root';
    private $password = "";
    private $database = "api_demo";

    public function getConnection(){
        $conn = new mysqli($this->host, $this->user, $this->password, $this->database);
        if($conn->connect_error){
            die("Error failed to connect to MySQL: " . $conn->connect_error);
        }
        else {
            return $conn;
        }
    }
}

?>
```

REST-API: Read(GET)

items/read.php

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

include_once '../config/Database.php';
include_once '../class/Items.php';

$database = new Database();
$db = $database->getConnection();

$items = new Items($db);

$items->id = (isset($_GET['id']) && $_GET['id']) ? $_GET['id'] :
'0';

$result = $items->read();
```

```
if($result->num_rows > 0){
    $itemRecords=array();
    $itemRecords["items"]=array();
        while ($item = $result->fetch_assoc()) {
            extract($item);
            $itemDetails=array(
                "id" => $id,
                "name" => $name,
                "description" => $description,
                    "price" => $price,
                "category_id" => $category_id,
                    "created" => $created,
                "modified" => $modified
            );
            array_push($itemRecords["items"], $itemDetails);
        }
    http_response_code(200);
    echo json_encode($itemRecords);
}else{
    http_response_code(404);
    echo json_encode(
        array("message" => "No item found.")
    );
}
```

REST-API: Read(GET)

Overview GET http://localhost/create + ... No Environment

http://localhost/create-simple-rest-api-php-mysql/items/read Save Send

GET http://localhost/create-simple-rest-api-php-mysql/items/read

Params Authorization Headers (8) Body • Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 14 ms Size: 3.47 KB Save as Example ...

Pretty Raw Preview Visualize JSON ...

```
1  {
2    "items": [
3      {
4        "id": 1,
5        "name": "ButterFlow Pen",
6        "description": "its... best pen",
7        "price": 900,
8        "category_id": 1,
9        "created": "2023-05-23 14:26:51",
10       "modified": "2014-05-31 17:42:26"
11      },
12      {
13        "id": 2,
14        "name": "Google Nexus 4",
15        "description": "The most awesome phone of 2013!",
16        "price": 299,
17        "category_id": 2,
18        "created": "2014-06-01 01:12:26",
19        "modified": "2014-05-31 17:42:26"
20      }
21    ]
22  }
```

REST-API: Create(POST)

items/create.php

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Max-Age: 3600");
header("Access-Control-Allow-Headers: Content-Type,
Access-Control-Allow-Headers, Authorization,
X-Requested-With");

include_once '../config/Database.php';
include_once '../class/Items.php';

$databse = new Database();
$db = $databse->getConnection();

$items = new Items($db);

$data = json_decode(file_get_contents("php://input"));
```

```
if(!empty($data->name) && !empty($data->description) &&
!empty($data->price) && !empty($data->category_id) &&
!empty($data->created)){
    $items->name = $data->name;
    $items->description = $data->description;
    $items->price = $data->price;
    $items->category_id = $data->category_id;
    $items->created = date('Y-m-d H:i:s');

    if($items->create()){
        http_response_code(201);
        echo json_encode(array("message" => "Item was created."));
    } else{
        http_response_code(503);
        echo json_encode(array("message" => "Unable to create
item."));
    }
} else{
    http_response_code(400);
    echo json_encode(array("message" => "Unable to create item.
Data is incomplete."));
}
?>
```

REST-API: Create(POST)

The screenshot displays a REST client interface with the following components:

- Overview Tab:** Shows the request method as **POST** and the URL as `http://localhost/create-simple-rest-api-php-mysql/items/create`. The environment is set to **No Environment**.
- Request Body:** The **Body** tab is selected, showing a JSON payload:

```
1 {
2   "name": "Gel Pen",
3   "description": "its best Pen",
4   "price": "90",
5   "category_id": "6",
6   "created": "2023-11-09 04:30:00"
7 }
```
- Response Body:** The **Body** tab is selected, showing a JSON response:

```
1 {
2   "message": "Item was created."
3 }
```
- Status Bar:** Indicates **Status: 201 Created**, **Time: 31 ms**, and **Size: 503 B**. It also includes a **Save as Example** button.

REST-API: Update(PUT)

items/update.php

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Max-Age: 3600");
header("Access-Control-Allow-Headers: Content-Type,
Access-Control-Allow-Headers, Authorization,
X-Requested-With");

include_once '../config/Database.php';
include_once '../class/Items.php';

$database = new Database();
$db = $database->getConnection();

$items = new Items($db);

$data = json_decode(file_get_contents("php://input"));
```

```
if(!empty($data->id) && !empty($data->name) &&
!empty($data->description) && !empty($data->price) &&
!empty($data->category_id)){
```

```
    $items->id = $data->id;
    $items->name = $data->name;
    $items->description = $data->description;
    $items->price = $data->price;
    $items->category_id = $data->category_id;
    $items->created = date('Y-m-d H:i:s');

    if($items->update()){
        http_response_code(200);
        echo json_encode(array("message" => "Item was
updated."));
    }else{
        http_response_code(503);
        echo json_encode(array("message" => "Unable to
update items."));
    }

} else {
    http_response_code(400);
    echo json_encode(array("message" => "Unable to update items.
Data is incomplete."));
}
?>
```

REST-API: Update(PUT)

The screenshot displays a REST client interface with the following components:

- Overview Tab:** Shows the request method as **PUT** and the URL as `http://localhost/create`.
- Request URL:** `http://localhost/create-simple-rest-api-php-mysql/items/update`
- Request Body:** A JSON object:

```
{
  "id": "61",
  "name": "Gel Pen",
  "description": "its best Pen",
  "price": "90",
  "category_id": "6",
  "created": "2023-11-09 04:30:00"
}
```
- Response Body:** A JSON object:

```
{
  "message": "Item was updated."
}
```
- Status:** 200 OK, Time: 6 ms, Size: 498 B

REST-API: Delete(DELETE)

items/delete.php

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Max-Age: 3600");
header("Access-Control-Allow-Headers: Content-Type,
Access-Control-Allow-Headers, Authorization,
X-Requested-With");

include_once '../config/Database.php';
include_once '../class/Items.php';

$database = new Database();
$db = $database->getConnection();

$items = new Items($db);

$data = json_decode(file_get_contents("php://input"));
```

```
if(!empty($data->id)) {
    $items->id = $data->id;
    if($items->delete()){
        http_response_code(200);
        echo json_encode(array("message" => "Item was
deleted."));
    } else {
        http_response_code(503);
        echo json_encode(array("message" => "Unable to
delete item."));
    }
} else {
    http_response_code(400);
    echo json_encode(array("message" => "Unable to delete items.
Data is incomplete."));
}
?>
```

REST-API: Delete(DELETE)

The screenshot displays a REST client interface with the following components:

- Overview Tab:** Shows the request method as **DEL** and the URL as `http://localhost/create-simple-rest-api-php-mysql/items/delete`.
- Request Configuration:** The **DELETE** method is selected in the dropdown. The URL is `http://localhost/create-simple-rest-api-php-mysql/items/delete`. The **Send** button is visible.
- Params, Authorization, Headers (8), Body, Pre-request Script, Tests, Settings:** The **Body** tab is active.
- Body Format:** The **Text** format is selected.
- Request Body:**

```
1 {  
2   "id": 61  
3 }
```
- Response Section:** The **Body** tab is active, showing the response in **Pretty** format.
 - Status:** 200 OK
 - Time:** 7 ms
 - Size:** 498 B
 - Save as Example** button is present.
- Response Body:**

```
1 {  
2   "message": "Item was deleted."  
3 }
```

