

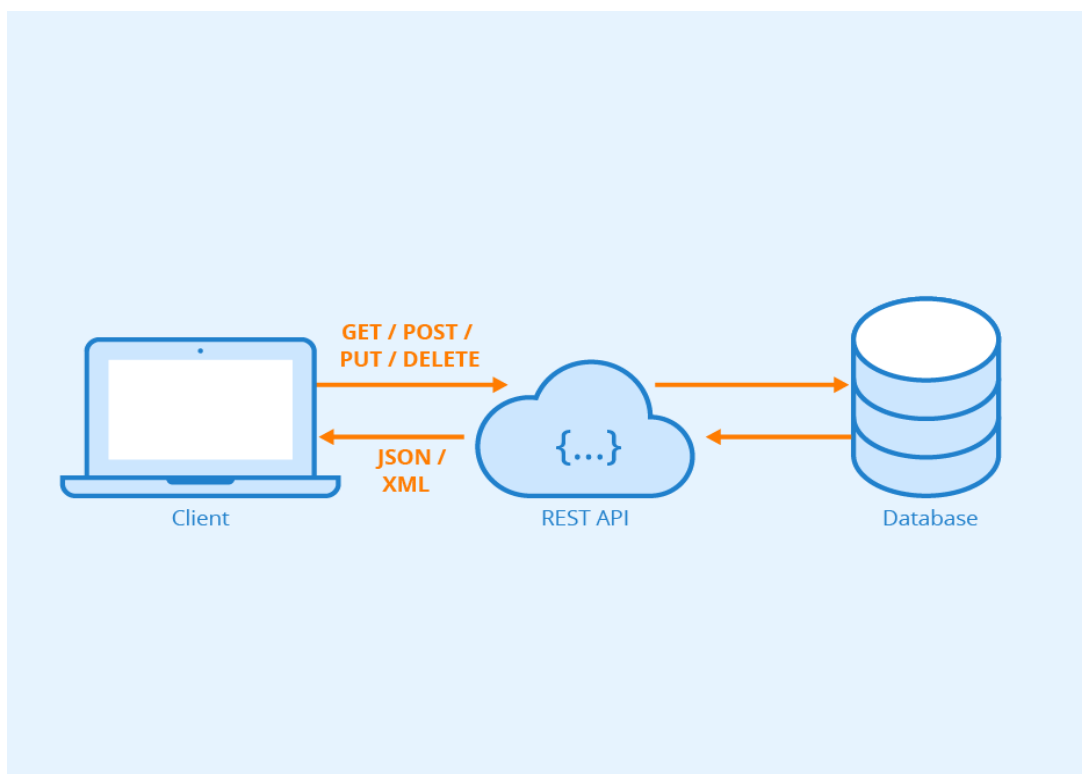
عنوان مقاله: آموزش API در زبان PHP

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: تیرماه ۱۴۰۱

منبع: <https://nikamooz.com/api-in-php/>

در صورتی که شما هم در دنیای برنامه‌نویسی فعالیتی داشته باشید، ممکن است نام API به گوشتان خورده باشد. اما آیا می‌دانید که این مفهوم به چه معناست؟ باید بدانید که API و Rest API ها به‌عنوان ستون فقرات توسعه وب به شمار می‌روند. اکثر برنامه‌های کاربردی در سطح وب که به‌عنوان برنامه‌های کاربردی فرانت‌اند توسعه داده می‌شوند از API های پشتیبانی شده و متصل از زبان‌های مختلف استفاده می‌کنند. ما در این مقاله قصد داریم تا در خصوص آموزش API در PHP به‌عنوان یکی از زبان‌های برنامه‌نویسی مهم، صحبت کنیم.

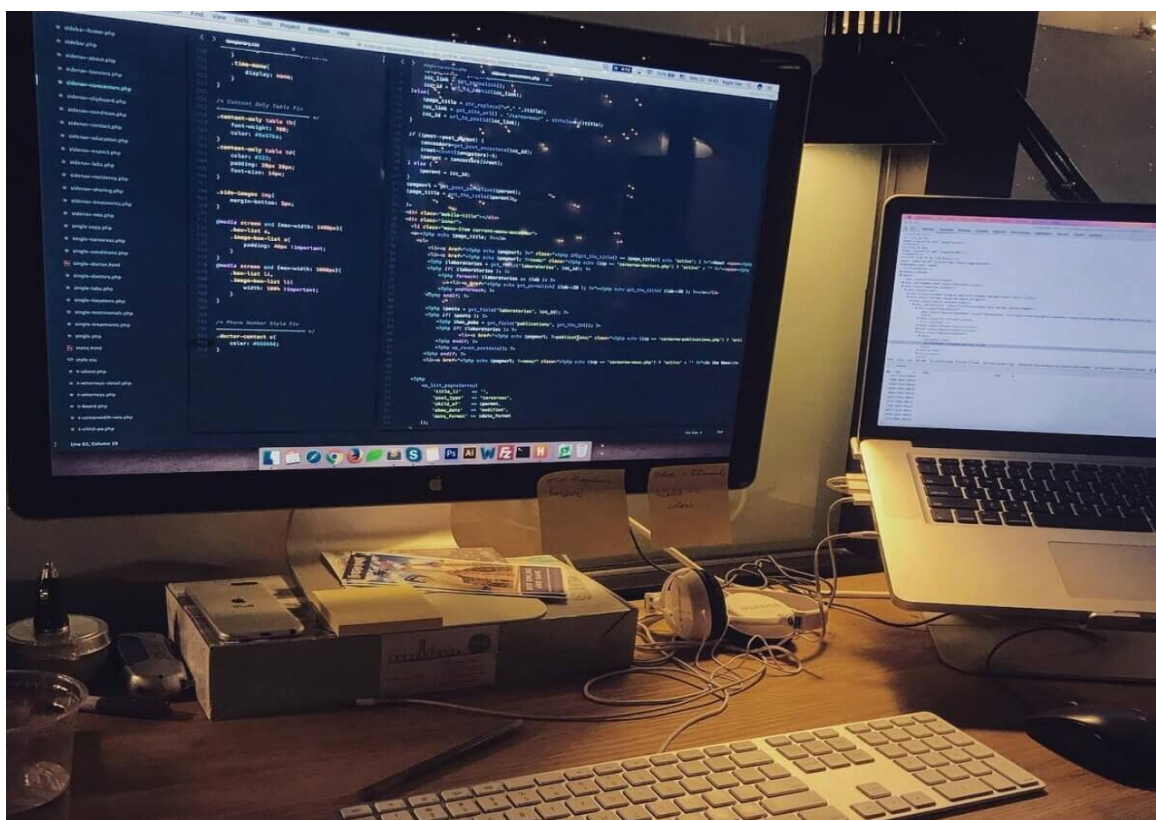


API مجموعه قوانینی برای ایجاد یک ارتباط مؤثر بین نرم‌افزارهای مختلف است

API چیست؟

پیش از اینکه در خصوص آموزش API در PHP صحبت کنیم، لازم است تا شما را با مفهوم API آشنا کنیم. API مخفف "Application Programming Interface" است به یک سری مجموعه قوانین اشاره دارد. این مجموعه قوانین به نرم افزار اجازه می دهند تا با یک نرم افزار دیگر ارتباط برقرار کنند. باید اشاره کرد که این قوانین می توانند شامل ایجاد، خواندن، به روزرسانی و یا حذف باشند. اما شاید بپرسید که ضرورت آموزش API در PHP چیست؟

باید بدانید که استفاده از API می تواند به عنوان ساده ترین راه برای ایجاد، خواندن، به روزرسانی و یا حذف اطلاعات برنامه از طریق اینترنت یا پروتکل Http شناخته شود. چنین اطلاعاتی می توانند به سرعت در اختیار کاربران قرار بگیرند. یکی دیگر از دلایل اصلی آموزش API در PHP این است که می توان از آن در هر برنامه ای که به اینترنت متصل می شود، استفاده کرد. اگر در هر برنامه ای مشاهده کردید که داده ها آن توسط یک برنامه دیگر ایجاد، به روزرسانی و یا حذف شده، بدانید که این عمل با کمک Apiها رخ داده است.



استفاده از API در بین برنامه نویسان زبان PHP رواج زیادی دارد

آموزش API در PHP:

در این قسمت از مقاله قصد داریم تا در خصوص آموزش RESTFUL API در PHP و API در PHP صحبت کنیم. برای اینکه آموزش API در PHP به صورت کامل در اختیار شما قرار بگیرد، ما به صورت بخش بخش و گام به گام در این خصوص صحبت خواهیم کرد؛ از این رو آموزش API در PHP به صورت زیر خواهد بود:

- اسکلت پروژه PHP را برای API خود ایجاد کنید:

اولین قدم در آموزش RESTFUL API در PHP این است که یک اسکلت برای پروژه خود ایجاد کنید. در این آموزش API در PHP ما از یک دایرکتوری src/ و یک فایل composer.json در دایرکتوری با یک وابستگی کار خود را آغاز خواهیم کرد. در این مرحله نیاز است تا کتابخانه DotEnv را که به ما این امکان را می‌دهد جزئیات احراز هویت Okta خود را در یک فایل env. خارج از برنامه نگهداری کنیم، نیاز خواهیم داشت.

```
composer.json

}

" require} : "

"    vlucas/phpdotenv": "^2.4"

,{

"    autoload} : "

"    psr-4} : "

"    Src||": "src"/

{

{
```

به کمک این قطعه کد در آموزش API در PHP ما یک بارگذاری خودکار PSR-4 را پیگیربندی کرده‌ایم که به صورت خودکار کلاس‌های PHP را در فهرست src/ جستجو می‌کند. همچنین ما یک دایرکتوری vendor / داریم و وابستگی DotEnv بر روی آن نصب شده است. همچنین در این آموزش API در PHP ما به یک فایل bootstrap.php نیاز داریم که متغیرهای محیطی ما را بارگیری کند.

```
<?php

require 'vendor/autoload.php';

use Dotenv\Dotenv;
```

```
$dotenv = new DotEnv(__DIR__);

$dotenv->load();

// test code, should output:

// api://default

// when you run $ php bootstrap.php

echo getenv('OKTAAUDIENCE');
```

- یک پایگاه داده برای API خود پیکربندی کنید:

ما در این آموزش API در PHP از MySQL برای تقویت API خود استفاده خواهیم کرد؛ از این رو نیاز است یک پایگاه داده و کاربر جدید برای برنامه خود ایجاد کنیم:

```
mysql -uroot -p

CREATE DATABASE api_example CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

CREATE USER 'api_user'@'localhost' identified by 'api_password';

GRANT ALL on api_example.* to 'api_user'@'localhost';

quit
```

مابقی API ما در این آموزش، تنها با یک موجودیت سروکار دارند: Person. اینم موجودیت دارای فیلدهای id، firstname, lastname, firstparent_id, secondparent_id خواهد بود. این موجودی به ما این امکان را می دهد تا افراد را با حداکثر دو والدین برای هر فرد تعریف کنیم؛ از این رو نیاز است تا جدول پایگاه داده خود را در MySQL ایجاد کنید.

- یک کلاس دروازه (Gateway) برای جدول افراد اضافه کنید

قدم بعدی در آموزش API در PHP ما، به این نیاز دارید تا یک کلاس دروازه (Gateway) برای جدول افراد اضافه کنید. الگوهای زیادی برای کار با پایگاه داده در زمینه شیء‌گرا وجود دارد. در این آموزش، ما از یک API ساده استفاده کرده‌ایم. از این رو طبیعی است که از یک الگوی ساده نیز بهره ببریم؛ به این دلیل تنها یک Table Gateway اجرا می‌کنیم.

```
<?php

namespace Src\TableGateways;

class PersonGateway {

    private $db = null;

    public function __construct($db)

    {

        $this->db = $db;

    }

    public function findAll()

    {

        $statement = "

            SELECT

                id, firstname, lastname, firstparent_id, secondparent_id

            FROM

                person;

        ";

        try {
```

```
$statement = $this->db->query($statement);

$result = $statement->fetchAll(\PDO::FETCH_ASSOC);

return $result;

} catch (\PDOException $e) {

    exit($e->getMessage());

}

}

public function find($id)

{

    $statement = "

        SELECT

            id, firstname, lastname, firstparent_id, secondparent_id

        FROM

            person

        WHERE id = ?;

    ";

    try {

        $statement = $this->db->prepare($statement);

        $statement->execute(array($id));

        $result = $statement->fetchAll(\PDO::FETCH_ASSOC);
```

```
return $result;

} catch (\PDOException $e) {

    exit($e->getMessage());

}

}

public function insert(Array $input)

{

    $statement = "

        INSERT INTO person

            (firstname, lastname, firstparent_id, secondparent_id)

        VALUES

            (:firstname, :lastname, :firstparent_id, :secondparent_id);

    ";

    try {

        $statement = $this->db->prepare($statement);

        $statement->execute(array(

            'firstname' => $input['firstname'],

            'lastname' => $input['lastname'],

            'firstparent_id' => $input['firstparent_id'] ?? null,

            'secondparent_id' => $input['secondparent_id'] ?? null,
```

```
));

return $statement->rowCount();

} catch (\PDOException $e) {

    exit($e->getMessage());

}

}

public function update($id, Array $input)

{

    $statement = "

    UPDATE person

    SET

        firstname = :firstname,

        lastname = :lastname,

        firstparent_id = :firstparent_id,

        secondparent_id = :secondparent_id

    WHERE id = :id;

    try {

        $statement = $this->db->prepare($statement);

        $statement->execute(array(

            'id' => (int) $id,
```



```
'firstname' => $input['firstname'],

'lastname' => $input['lastname'],

'firstparent_id' => $input['firstparent_id'] ?? null,

'secondparent_id' => $input['secondparent_id'] ?? null,

));

return $statement->rowCount();

} catch (\PDOException $e) {

    exit($e->getMessage());

}

}

public function delete($id)

{

    $statement = "

        DELETE FROM person

        WHERE id = :id;

    ";

    try {

        $statement = $this->db->prepare($statement);

        $statement->execute(array('id' => $id));

        return $statement->rowCount();

    }

}
```

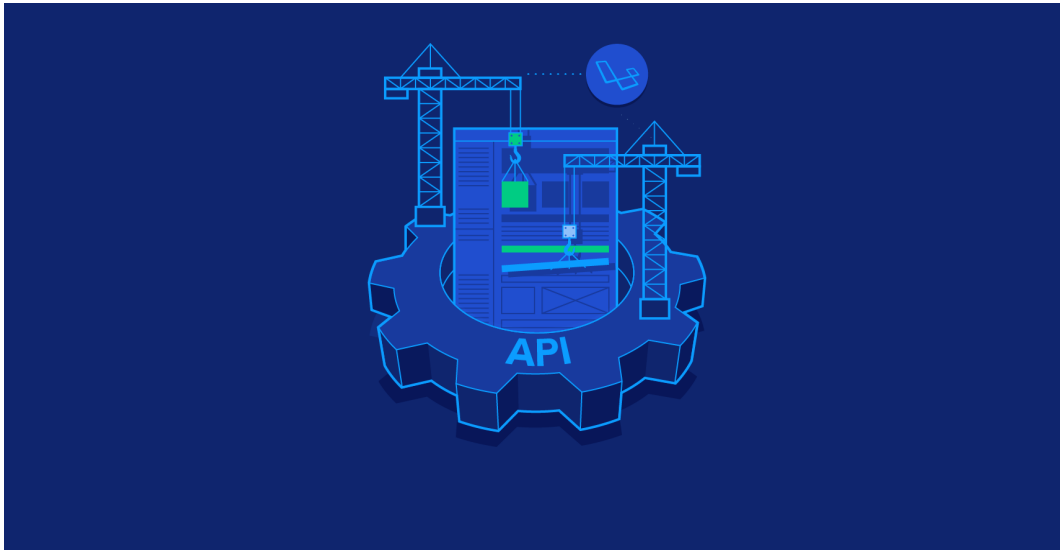
```

} catch (\PDOException $e) {

    exit($e->getMessage());

}

}
    
```



پیاده‌سازی API در PHP مزایای بسیار زیادی دارد

• PHP API را پیاده‌سازی کنید:

یکی از مهم‌ترین قسمت‌ها در آموزش API در PHP پیاده‌سازی آن خواهد بود. ما یک فایل `public/index.php/` ایجاد خواهیم کرد تا به‌عنوان کنترلر عمل کرده و درخواست‌ها را پردازش کند. همچنین یک `src/Controller/PersonController.php` برای مدیریت نقاط پایانی API ایجاد خواهیم کرد.

```

<?php

require "../bootstrap.php";

use Src\Controller\PersonController;

header("Access-Control-Allow-Origin: *");

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

header("Access-Control-Allow-Methods: OPTIONS,GET,POST,PUT,DELETE");
    
```

```
header("Access-Control-Max-Age: 3600");

header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-
Requested-With");

$uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);

$uri = explode('/', $uri);

// all of our endpoints start with /person

// everything else results in a 404 Not Found

if ($uri[1] !== 'person') {

    header("HTTP/1.1 404 Not Found");

    exit();

}

// the user id is, of course, optional and must be a number:

$userId = null;

if (isset($uri[2])) {

    $userId = (int) $uri[2];

}

$requestMethod = $_SERVER["REQUEST_METHOD"];

// pass the request method and user ID to the PersonController and process the HTTP request:

$controller = new PersonController($dbConnection, $requestMethod, $userId);

$controller->processRequest();
```

• API PHP خود را با OAuth 2.0 ایمن کنید

در قدم آخر آموزش API در PHP نیاز است تا API PHP خود را با OAuth 2.0 ایمن کنید. ما در این آموزش RESTFUL API در PHP از okta به عنوان سرور مجوز و برای ایجاد جریان اعتبار مشتری استفاده می‌کنیم. این جریان برای احراز هویت توصیه می‌شود و به هر مشتری یک شناسه و رمز می‌دهد.

نتیجه‌گیری:

آموزش RESTFUL API در PHP طرفداران بسیار زیادی پیدا کرده است. از apiها می‌توانید در تمامی برنامه‌های کاربردی که به اینترنت متصل می‌شوند، استفاده کنید. این قوانین به نرم‌افزارهای اجازه می‌دهد تا با سایر نرم‌افزارها ارتباط برقرار کنند. ما در این مقاله در خصوص آموزش api در php صحبت کردیم و نکات مهمی را در این زمینه با شما به اشتراک گذاشتیم.

سؤالات متداول

ما در این قسمت قصد داریم تا به پرتکرارترین و متداول‌ترین سؤالات شما در زمینه آموزش api در php پاسخ دهیم.

۱- آیا می‌توانیم از PHP برای API استفاده کنیم؟

بله؛ شما می‌توانید در زبان برنامه‌نویسی از فریم‌ورک‌های بسیار زیادی برای ساخت api استفاده کنید.

۲- API در PHP چیست؟

یک رابط برنامه‌نویسی یا api، کلاس‌ها، توابع، متدها و روش‌هایی را تعریف می‌کند که برنامه شما برای انجام وظایف خود باید آن‌ها را فراخوانی کند.

۳- RESTFUL API در PHP چیست؟

REST (Representational State Transfer) یک api است که مجموعه‌ای از توابع را تعریف می‌کند که برنامه‌نویسان می‌توانند از آن‌ها برای ارسال درخواست و یا دریافت پاسخ استفاده کنند.

۴- آیا یادگیری REST API دشوار است؟

از آن جایی که rest از روش‌های Http برای ارتباط استفاده می‌کند و اکثر برنامه‌نویسان با Http آشنا هستند، یادگیری آن دشوار نخواهد بود.

۵- چند نوع API در PHP وجود دارد؟

چهار نوع اصلی API وجود دارد که در برنامه‌های کاربردی مبتنی بر وب استفاده می‌شوند: public, partner, private و composite .