

WP REST API Einführung

16.12.2015

Martin Sotirov

Inhalt

- Definition
- Infrastruktur
- Endpoints
- Beispiele

RESTful Web

- Representational State Transfer
- Im Jahr 2000 von Roy Fielding eingeführt
- Abstraktion der Struktur und des Verhaltens einer Webseite

`GET http://example.com/authors/stephen-king`

Create Read Update Delete

- Create = POST
- Read = GET
- Update = PUT
- Delete = DELETE

Operationen eines RESTful APIs

Inhalt	GET	PUT	POST	DELETE
Kollektion <i>/authors/</i>	Kollektion lesen	Ganze Kollektion aktualisieren	Neues Element erstellen	Kollektion löschen
Ein Element <i>/authors/stephen-king</i>	Einziges Element lesen	Einziges Element aktualisieren		Einziges Element löschen

Auch WordPress wird RESTful

- 2013 von Ryan McCue als Plugin programmiert
- WordPress 4.4 hat schon die REST-Infrastruktur
- Die Endpoints kommen mit WP 4.5
- <https://wordpress.org/plugins/rest-api/>

REST Infrastruktur

```
class WP_HTTP_Server { ... }
```

```
class WP_HTTP_Request { ... }
```

```
class WP_HTTP_Response { ... }
```

Klassen unter `/wp-includes/rest-api/` zu finden

Hooks und Funktionen unter `/wp-includes/rest-api.php`

Eigene Endpoints erstellen

```
function rp_register_endpoints() {  
    $controller = new WP_REST_Plugins_Controller();  
    $controller->register_routes();  
}  
add_action( 'rest_api_init', 'rp_register_endpoints' );
```


Eigene Endpoints erstellen

```
register_rest_route( $path, $methods );
```

Eigene Endpoints erstellen

```
$namespace = 'rp/v1';  
$base = 'plugins';  
  
register_rest_route( $namespace, '/' . $base, array(  
    array(  
        'methods'           => WP_REST_Server::READABLE,  
        'callback'          => array( $this, 'get_items' ),  
        'permission_callback' => array( $this, 'permissions_check' ),  
        'args'              => array(),  
    ),  
) );
```

Eigene Endpoints erstellen

```
public function get_items( $request ) {  
    $data = get_awesome_data();  
    return new WP_REST_Response( $data );  
}
```

```
public function permissions_check() {  
    if (current_user_can( 'manage_options' )) {  
        return true;  
    }  
}
```

Core Endpoints

`/wp/v2/posts/`

`/wp/v2/posts/59/`

`/wp/v2/pages/`

`/wp/v2/media/`

`/wp/v2/taxonomies/`

`/wp/v2/terms/category/`

CPT Endpoints

```
register_post_type( 'author', array(  
    ...  
    'show_in_rest'           => true,  
    'rest_base'              => 'authors',  
    'rest_controller_class' => 'WP_REST_Posts_Controller',  
) );
```

</wp/v2/authors/>

Custom Fields

```
register_api_field(  
    array( 'author' ), // custom post type  
    'bio',             // custom field  
    array(  
        'get_callback'    => 'get_field_bio',  
        'update_callback' => 'update_field_bio',  
        'schema'          => null,  
    )  
);
```

/wp/v2/authors/

Authentifizierung

- Generell nur GET-Operationen ohne Login
- Authentifizierung über [Basic Auth](#), [Cookie](#) oder [OAuth \(v1\)](#)

Authentifizierung

Post lesen	Posts lesen	Post aktualisieren	Post erstellen
Public Post Meta lesen	Post Meta aktualisieren	Private Post Meta lesen	User erstellen
Kommentare lesen	Kommentare erstellen	Kommentare löschen	User aktualisieren

Authentifizierungsmethoden

- Authorization: Basic QWxhZGRpbjpvYVUHNlc2FtZQ
Base64 kodiert username:password
<https://wordpress.org/plugins/wp-basic-auth/>
- X-WP-Nonce: 5b5c8435f2
Mit `wp_create_nonce('wp_rest')` erstellen
- OAuth v1
<https://github.com/WP-API/OAuth1/>

Beispiele

Fragen