



با سلام

دو روز پیش وردپرس ۴.۴ با عنوان clifford منتشر شد. که ویژگی های جالبی رو داشت مثلا ویژگی embed که شما اگه خواستین از مطلب دیگه سایتتون تو مطلب دیگه رفرنس بدین و خواستین خلاصه ای از اونم هم نشون بدین کافیه آدرس URL مطلب دیگه رو تو ویرایشگرتون paste کنین و اون رو بصورت خلاصه پست ببینین. مثلا برای قسمت زیر من فقط آدرس پست رو گذاشتم:

[استفاده از Notification در وب با جاوااسکریپت](#)

و ویژگی مهم و عالی دیگه Rest api هست که از نظر من فوق العاده س. البته این ویژگی مربوط میشه به افزونه [WP REST API](#) که وردپرس اومده این افزونه رو با هسته خودش تلفیق کرده و به توسعه دهنده ها اجازه میده تا هر مدل api که میخوان برا افزونه شون بنویسن و دیگه نیازه به نصب افزونه WP REST API نباشه. مثلا برا سایتشون یک اپلیکیشن اندروید با حتی دسکتاپ بنویسن.

استفاده از ویژگی جدید وردپرس ۴.۴ به نام rest api

افزونه WP REST API بطور پیشفرض به سری api ها داشت مثل گرفتن پست ها، پست، دیدگاه ها و ... ولی در وردپرس ۴.۴ فقط به مسیر یا route براش مشخص شده که میتونین با تابع [get_rest_url](#) آدرس رو در بیارین و ببینین که بطور پیشفرض مقدارش برابره با <http://yoursite.com/wp-json/> هست. ولی خب این میتونه با فیلتر [rest_url](#) تغییر کنه. نتیجه میشه تصویر زیر یا اینکه از [اینجا](#) ببینین:

```

{
  "name": "مرجع کدهای برنامه نویسی وب",
  "description": "مرجع کدهای برنامه نویسی وب - PHP,wordpress",
  "url": "http://ircodex.ir",
  "namespaces": [
    "oembed/1.0"
  ],
  "authentication": [],
  "routes": {
    "/": {
      "namespace": "",
      "methods": [
        "GET"
      ],
      "endpoints": [
        {
          "methods": [
            "GET"
          ],
          "args": {
            "context": {
              "required": false,
              "default": "view"
            }
          }
        }
      ]
    }
  }
}

```



```

24
25
26  /**
27   * Alias for DELETE transport method.
28   *
29   * @since 4.4.0
30   * @var string
31   */
32  const DELETABLE = 'DELETE';
33
34  /**
35   * Alias for GET, POST, PUT, PATCH & DELETE transport methods together.
36   *
37   * @since 4.4.0
38   * @var string
39   */
40  const ALLMETHODS = 'GET, POST, PUT, PATCH, DELETE';
  
```

و callback هم مشخص میکنه وقتی به این مسیر هداست شد چه اتفاقی بیفته که خب به تابع براش تعریف کردیم. خیلی ساده تو این تابع گفتیم که تمام کاربران ما رو بگیر و با استفاده از کلاس WP_REST_Response به پاسخ یا response استاندارد ساختیم و اون رو return کردیم. خودتون انجام بدین نتیجه ش جالبه.

ارسال اطلاعات از طریق rest api

اولین نکته تو ارسال اطلاعات متد ارسال اطلاعات هست. پس اگه ما بخوایم به user_id از طریق متد GET بفرستیم.

```

1  add_action('rest_api_init', function(){
2      register_rest_route('ircodex_api', 'users', array(
3          'methods' => WP_REST_Server::READABLE,
4          'callback' => 'ircodex_users_callback'
5      ));
6
7      register_rest_route('ircodex_api', 'user', array(
8          'methods' => WP_REST_Server::READABLE,
9          'callback' => 'ircodex_user_callback',
10         'args' => array(
11             'user_id' => array(
12                 'required' => true,
13                 'sanitize_callback' => 'absint'
14             )
15         )
16     ));
17
18 });
19
20 function ircodex_users_callback(){
21     $response = new WP_REST_Response(get_users());
22     return $response;
23 }
24
25 function ircodex_user_callback($request){
26     $response = new WP_REST_Response(get_users(array(
27         'include' => array($request['user_id'])
28     )));
29     return $response;
30 }
  
```

حالا اگه آدرس http://yoursite.com/wp-json/ircodex_api/user?user_id=1 رو بزنین میبینین که فقط اطلاعات کاربر با شناسه ۱ براتون میاد. و خب با تغییر post_id میتونین تصمیم بگیرید که کدوم کاربر رو نشون بده.

خط ۱۰ - مشخص کردیم که میخوایم آرگومانهایی هم ارسال کنیم برا تغییر نتیجه.

خط ۱۱ - عنوان اولین آرگومان ما user_id رو مشخص کردیم یعنی با `$_GET['user_id']` دسترسیه.

خط ۱۲ - گفتیم که این داده برای این مسیر الزامیه اگه شما پسوند user_id=1 رو بزنین خطوط زیر لراتون نشون داده میشه:

```

1  {
2      "code": "rest_missing_callback_param",
3      "message": "Missing parameter(s): user_id",
4      "data": {
5          "status": 400,
6          "params": [
7              "user_id"
8          ]
9      }
10 }
  
```

خط ۱۳ - می‌گیم مقدار ورودی با این تابع فیلتر یا تصحیح بشه که خب چون شناسه کاربر فقط عدد صحیحیه از تابع absint استفاده کردیم.
 خط ۲۷ - دیگه ما با متغیر GET کاری نداریم چون خوش درون آرگومان ورودی تابع فیلتر شده و تحویل ما داده میشه و ما هم با توجه به شناسه کاربر مورد نظر رو نشون دادیم.
 حالا اگه خواستی اطلاعات رو از طریق متد پست بفرستی میتونین دو کار انجام بدین اول کاری کنین که اطلاعات فقط از طریق POST قابل دریافت باشه و یا اینکه علاوه بر GET بشه از طریق POST هم ارسال اطلاعات انجام داد. ولی من ترجیح میدم فقط با POST ارسال بشه.

```

1  add_action('rest_api_init', function(){
2      register_rest_route('ircodex_api', 'users', array(
3          'methods' => WP_REST_Server::READABLE,
4          'callback' => 'ircodex_users_callback'
5      ));
6
7      register_rest_route('ircodex_api', 'user', array(
8          'methods' => WP_REST_Server::READABLE,
9          'callback' => 'ircodex_user_callback',
10         'args' => array(
11             'user_id' => array(
12                 'required' => true,
13                 'sanitize_callback' => 'absint'
14             )
15         )
16     ));
17
18     register_rest_route('ircodex_api', 'post', array(
19         'methods' => WP_REST_Server::CREATABLE,
20         'callback' => 'ircodexpost_callback',
21         'args' => array(
22             'user_id' => array(
23                 'default' => 347,
24                 'sanitize_callback' => 'absint'
25             )
26         )
27     ));
28 });
29
30 function ircodex_users_callback(){
31     $response = new WP_REST_Response(get_users());
32     return $response;
33 }
34
35 function ircodex_user_callback($request){
36     $response = new WP_REST_Response(get_users(array(
37         'include' => array($request['post_id'])
38     )));
39     return $response;
40 }
41
42 function ircodexpost_callback($request){
43     $response = new WP_REST_Response(get_post($request['user_id']));
44     return $response;
45 }
46 }
  
```

اومدیم یه مسیر که فقط از طریق POST در دسترسه ایجاد کردیم. خط ۱۹ اومدیم از WP_REST_Server::CREATABLE استفاده کردیم تا متد POST رو مشخص کرده باشیم. (البته معمولا برای ایجاد پست و این چیزا از این متد استفاده میکنن برا همین CREATABLE هست ولی خب ما اینجا برا آموزش برا گرفتیم پست ازش استفاده میکنیم) خط ۲۳ اجباری بودن پارامتر رو برداشتیم و گفتیم اگه فرستاده نشه مقدار ۳۴۷ رو به عنوان پیشفرض قرار بده. محتوای تابع هم که کاملا واضحه.

استفاده از عبارات با قاعده در rest api

حالا اگه بخواین به صورت wild card استفاده کنین از این چطور میشه. مثلا شما میخواین با این ساختار url ارتباط برقرار کنین:
http://yoursite.com/wp-json/ircodex_api/author/{author_id}

```

1  add_action('rest_api_init', function(){
2      register_rest_route('ircodex_api', 'users', array(
3          'methods' => WP_REST_Server::READABLE,
4          'callback' => 'ircodex_users_callback'
5      ));
6
7      register_rest_route('ircodex_api', 'user', array(
8          'methods' => WP_REST_Server::READABLE,
9          'callback' => 'ircodex_user_callback',
10         'args' => array(
11             'user_id' => array(
12                 'required' => true,
13                 'sanitize_callback' => 'absint'
14             )
15         )
16     ));
17
18     register_rest_route('ircodex_api', 'post', array(
19         'methods' => WP_REST_Server::CREATABLE,
20         'callback' => 'ircodexpost_callback',
21         'args' => array(
22             'user_id' => array(
23                 'default' => 347,
24                 'sanitize_callback' => 'absint'
25             )
26         )
27     ));
28
29     register_rest_route('ircodex_api', '/author/(?P<id>\d+)', array(
30         'methods' => WP_REST_Server::READABLE,
31         'callback' => 'ircodex_user_wild_callback',
32     ));
33
34 });
35
36 function ircodex_users_callback(){
37     $response = new WP_REST_Response(get_users());
38     return $response;
39 }
40
41 function ircodex_user_callback($request){
42     $response = new WP_REST_Response(get_users(array(
43         'include' => array($request['post_id'])
44     )));
45     return $response;
46 }
47
48 function ircodexpost_callback($request){
49     $response = new WP_REST_Response(get_post($request['user_id']));
50     return $response;
51 }
52
53 function ircodex_user_wild_callback($data){
54     return $data['id'];
55 }
    
```

نکته مهم تو خطوط ۲۹ و ۵۴ نهفته س. نیازی به توضیح نداره یعنی باید بدونین با این wildcard ها چطور باید کار کرد.
حالا به کم پیشرفته تر:

```

1  add_action('rest_api_init', function(){
2      register_rest_route('ircodex_api', 'users', array(
3          'methods' => WP_REST_Server::READABLE,
4          'callback' => 'ircodex_users_callback'
5      ));
6
7      register_rest_route('ircodex_api', 'user', array(
8          'methods' => WP_REST_Server::READABLE,
9          'callback' => 'ircodex_user_callback',
10         'args' => array(
11             'user_id' => array(
12                 'required' => true,
13                 'sanitize_callback' => 'absint'
14             )
15         )
16     ));
17
18     register_rest_route('ircodex_api', 'post', array(
19         'methods' => WP_REST_Server::CREATABLE,
20         'callback' => 'ircodexpost_callback',
    
```

```

21     'args' => array(
22         'user_id' => array(
23             'default' => 347,
24             'sanitize_callback' => 'absint'
25         )
26     )
27 );
28
29 register_rest_route('ircodex_api', '/author/(?P<id>\d+)', array(
30     'methods' => WP_REST_Server::READABLE,
31     'callback' => 'ircodex_user_wild_callback',
32 );
33
34 register_rest_route('ircodex_api', '/(?P<type>[a-z]+)/(?P<id>\d+)', array(
35     'methods' => WP_REST_Server::READABLE,
36     'callback' => 'ircodex_user_new_wild_callback',
37 );
38
39 });
40
41 function ircodex_users_callback(){
42     $response = new WP_REST_Response(get_users());
43     return $response;
44 }
45
46 function ircodex_user_callback($request){
47     $response = new WP_REST_Response(get_users(array(
48         'include' => array($request['post_id'])
49     )));
50     return $response;
51 }
52
53 function ircodexpost_callback($request){
54     $response = new WP_REST_Response(get_post($request['user_id']));
55     return $response;
56 }
57
58 function ircodex_user_wild_callback($data){
59     return $data['id'];
60 }
61
62 function ircodex_user_new_wild_callback($data){
63     return $data['type'] . ' => ' . $data['id'];
64 }

```

خطوط ۳۴ تا ۳۷ و ۶۲ تا ۶۴ جدیدین و با آزمایش همه چی دستتون میاد.
یعنی ساختار شما شده این:

http://yoursite.com/wp-json/ircodex_api/{type}/{author_id}

البته به همراه این ساختار همیشه از متد های POST و GET و حتی سایر متد ها هم استفاده کرد.
یه آموزش کوچیک و نقشه راه بود.

موفق باشد.

نظر بدین

خدانگهدار