

# Manual: Custom format for detail and apply url

Version 1.0





# Contents

Contents.....	1
1. General.....	2
2. Configuration .....	2
2.1 Activation .....	2
2.2 Configure slug format .....	2
2.3 Configure URL format .....	3
3. Usage .....	3
3.1 Generating the slug.....	3
3.2 Updating slug automatically .....	3
3.3 Updating slug manually .....	4
4. API calls .....	4
4.1 Retrieve vacancy list .....	4
4.2 Retrieve vacancy detail with uid.....	6
4.3 Retrieve vacancy detail with slug .....	8
4.4 Retrieve vacancy URLs .....	10
5. Other information.....	12



# 1. General

OTYS now supports the option to configure your own URL structure for detail and apply URLs. This should be used in combination with CMS partners. The feature is made with that in mind. The manual will go over the configuration in OTYS and how a partner can integrate it on their end.

## 2. Configuration

The configuration consists of a few steps which are described below.

### 2.1 Activation

Before the custom URL format can be used, OTYS needs to enable the functionality. If the settings in the next chapters are not visible, create a Support ticket asking if client setting 'Vacancies – Enable custom url slug-system' can be enabled.

***After activating this functionality, already published vacancies need to be updated manually to generate a slug. This is something we want to improve in a next iteration. For now, keep this in mind.***

### 2.2 Configure slug format

The slug is an integral part of the custom URL structure OTYS has created. By default the slug format is set to Title + UID of the vacancy. Via website setting 'Vacancies – Slug format' the slug format can be adjusted per website. The structure of the fields is based on our API. A field needs to be placed between curly brackets. Below fields are supported:

- ✔ VacancyService.uid
- ✔ VacancyService.internalId
- ✔ VacancyService.referenceNr
- ✔ VacancyService.externalReferenceNr
- ✔ VacancyService.title
- ✔ VacancyService.extraTitle
- ✔ VacancyService.category
- ✔ VacancyService.JobTypeName
- ✔ VacancyService.location

For example, you want to save the slug with *title* and *location*, you do this as:

***{{VacancyService.title}}-{{VacancyService.location}}***



## 2.3 Configure URL format

After configuring the slug, you want to configure the structure of the URL. You can do this per website and per language. This can be done via website setting 'Vacancy url format'. The url format can be configured as:

1. **Domain:** Fill in the domain of the website. For instance 'https://mywebsite.nl/' (Do not forget the forward slash)
2. **Detail slug:** Fill in what comes after the domain how the vacancy detail url should be. For instance 'vacancies/{{slug}}'. Do not forget the slug part. This is what is generated based upon setting 'Vacancies – Slug format'.
3. **Apply slug:** Fill in what comes after the domain how the vacancy apply url should be. For instance 'vacancies/{ {slug} }/apply'. Do not forget the slug part. This is what is generated based upon setting 'Vacancies – Slug format'.
4. After filling the three fields you can check how the complete url looks like in the preview at the bottom of the setting. If all looks good, click on 'Save' to apply the changes
5. If you have a multilingual website, change the language flag and follow the previous steps once more

*Note: The `{{slug}}` can be placed anywhere in the detail or apply field. Just keep in mind to add this variable, otherwise the url is generated without it.*

# 3. Usage

Now that the configuration is done, it is time to understand the logic of these custom URLs.

## 3.1 Generating the slug

By default the slug in the URL, is automatically generated based on website setting 'Website - Slug format'. If the slug is generated, it is sanitized of capital letters, spacing and special characters. Also a check is done to make sure it is unique. If not, the slug is incremented.

Examples based on format Title + UID:

- ✔ PHP Developer > php-developer-hfg44hcb29haa
- ✔ Web Developer needed in Amsterdam! > web-developer-needed-in-amsterdam- hfg44hcb29haa

## 3.2 Updating slug automatically

When you create a new empty vacancy, the vacancy title will be 'New job'. If the slug format is default, the slug of this vacancy will be 'new-job-[oid]'. If you change the vacancy title to for instance 'PHP Developer' it changes the slug to 'php-developer-[oid]'. Every time a field is updated that is part of the slug format, the slug is updated. The old slug, is still available in the database and can still be used to retrieve the vacancy information via the API.



### 3.3 Updating slug manually

As mentioned above, the slug is updating automatically when a field that is part of the slug format is updated. Sometimes you want to manually set the slug and keep it like that. This is possible if user setting 'Vacancies - Enable slug field' is enabled for the user. You can see the field here:

1. Open the Vacancy module
2. Open a vacancy
3. Click on 'More' and choose 'URL'
4. The modal with URLs opens. The first field shows the slug that is active

If you change the slug and save it, the slug is set to 'manual mode' in the database. This means that an update to the vacancy which includes one of the slug format fields, will not update the slug anymore, since you chose to manually set it. If you change your mind and want to re-enable the automatic mode, you can click on the 'reset button' on the right side of the slug field.

## 4. API calls

Below you can find a list of example API calls to get a better understanding of the requests and responses that involve slugs.

### 4.1 Retrieve vacancy list

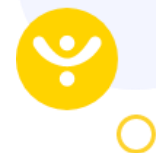
#### Request

Using method 'OtyS.Services.VacancyService.getList' you can retrieve a list of all vacancies.

```
{
  "jsonrpc": "2.0",
  "method": "OtyS.Services.VacancyService.getList",
  "params": [
    "[sessiontoken]"
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.



The request above will return all option fields. If you only need specific option fields, you can also specify them in your request:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getList",
  "params": [
    "[sessiontoken]",
    [
      "[fields]"
    ]
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [fields]: This is an array of the specific fields for which you would like to retrieve the options.

### Example request

Let's say you want to retrieve the options for fields 'uid', 'title' & 'slug'; then the request would be as follows:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getList",
  "params": [
    "[sessiontoken]",
    {
      "what": {
        "uid": 1,
        "title": 1,
        "slug": 1
      }
    }
  ],
  "id": 1
}
```



## Response

Based upon the request mentioned above, you will get the following response:

```
{
  "result": [
    {
      "uid": "[vacancy-uid]",
      "title": "[vacancy-title]",
      "slug": {
        "[siteid]": "[slug]"
      },
    },
  ],
  "id": 1
}
```

In this response above you see the following parameters:

- ✔ [siteid]: This is the id of the website
- ✔ [slug]: This is the name of the slug for that specific website

## 4.2 Retrieve vacancy detail with uid

### Request

Using method 'Otys.Services.VacancyService.getDetail' you can retrieve the detail of one vacancy.

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getDetail",
  "params": [
    "[sessiontoken]",
    "[vacancyuid]"
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [vacancyuid]: This is the uid of the vacancy

The request above will return all option fields. If you only need specific option fields, you can also specify them in your request:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getDetail",
  "params": [
    "[sessiontoken]",
    "[vacancyuid]",
    [
      "[fields]"
    ]
  ]
}
```



```
],  
  "id": 1  
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [vacancyuid]: This is the uid of the vacancy
- ✔ [fields]: This is an array of the specific fields for which you would like to retrieve the options.

### Example request

Let's say you want to retrieve the options for fields 'uid', 'title' & 'slug'; then the request would be as follows:

```
{  
  "jsonrpc": "2.0",  
  "method": "OtyS.Services.VacancyService.getDetail",  
  "params": [  
    "[sessiontoken]",  
    "[vacancyuid]",  
    {  
      "uid": 1,  
      "title": 1,  
      "slug": 1  
    }  
  ],  
  "id": 1  
}
```

### Response

Based upon the request mentioned above, you will get the following response:

```
{  
  "result": [  
    {  
      "uid": "[vacancy-uid]",  
      "title": "[vacancy-title]",  
      "slug": {  
        "[siteid]": "[slug]"  
      }  
    },  
  ],  
  "id": 1  
}
```

In this response above you see the following parameters:

- ✔ [siteid]: This is the id of the website
- ✔ [slug]: This is the name of the slug for that specific website





### 4.3 Retrieve vacancy detail with slug

#### Request

Using method 'Otys.Services.VacancyService.getDetailBySlug' you can retrieve the detail of one vacancy.

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getDetailBySlug",
  "params": [
    "[sessiontoken]",
    "[vacancyslug]"
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [vacancyslug]: This is the slug of the vacancy

The request above will return all option fields. If you only need specific option fields, you can also specify them in your request:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getDetailBySlug",
  "params": [
    "[sessiontoken]",
    "[vacancyslug]",
    [
      "[fields]"
    ]
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [vacancyslug]: This is the slug of the vacancy
- ✔ [fields]: This is an array of the specific fields for which you would like to retrieve the options.



### Example request

Let's say you want to retrieve the options for fields 'uid', 'title' & 'slug'; then the request would be as follows:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getDetailBySlug",
  "params": [
    "[sessiontoken]",
    "[vacancyslug]",
    {
      "uid": 1,
      "title": 1,
      "slug": 1
    }
  ],
  "id": 1
}
```

### Response

Based upon the request mentioned above, you will get the following response:

```
{
  "result": [
    {
      "uid": "[vacancy-uid]",
      "title": "[vacancy-title]",
      "slug": {
        "[siteid]": "[slug]"
      }
    },
  ],
  "id": 1
}
```

In this response above you see the following parameters:

- ✔ [siteid]: This is the id of the website
- ✔ [slug]: This is the name of the slug for that specific website



## 4.4 Retrieve vacancy URLs

### Request

Using method 'Otys.Services.VacancyService.getUrls' you can retrieve the detail of one vacancy.

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getUrls",
  "params": [
    "[sessiontoken]",
    "[vacancyuid]"
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [vacancyuid]: This is the uid of the vacancy

The request above will return all option fields. If you only need specific option fields, you can also specify them in your request:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getUrls",
  "params": [
    "[sessiontoken]",
    "[vacancyuid]",
    [
      "[fields]"
    ]
  ],
  "id": 1
}
```

In this request above you see the following parameters:

- ✔ [sessiontoken]: This is the session token used for authenticating (almost) all OWS requests.
- ✔ [vacancyuid]: This is the uid of the vacancy
- ✔ [fields]: This is an array of the specific fields for which you would like to retrieve the options.



## Example request

Let's say you want to retrieve the options for fields 'uid', 'title' & 'slug'; then the request would be as follows:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.VacancyService.getUrls",
  "params": [
    "[sessiontoken]",
    "[vacancyuid]",
    {
      "jobUrl": 1,
      "applyUrl": 1,
      "slug": 1,
      "slugAutoUpdate": 1
    }
  ],
  "id": 1
}
```

## Response

Based upon the request mentioned above, you will get the following response:

```
{
  "result": {
    "jobUrl": "[job-url]",
    "applyUrl": "[apply-url]",
    "slug": "[slug]",
    "slugAutoUpdate": "[slug-value]"
  },
  "id": 1
}
```

In this response above you see the following parameters:

- ✔ [slug-value]: This is 0 or 1. If the value is 0, this means the slug is manually set. If the value is 1, this means the slug will auto update if a parameter from the slug is updated. For instance the title of the vacancy.



## 5. Other information

This is the first version of custom URLs. If you find a bug or something does not work according to the description, please create a support ticket via your partner client. If you do not have such a client, then send an email to [partners@otys.nl](mailto:partners@otys.nl) with a detailed description of the problem and for which client you are integrating.

If you have ideas how we can improve this functionality further, we would love to hear from you! This can also be done via a support ticket or email us.