wagtail-personalisation Documentation

Release 1.0.4

Lab Digital BV

Mar 15, 2019

Contents:

CHAPTER 1

Getting started

1.1 Installing Wagxperience

1.1.1 Installing the module

The Wagxperience app runs in the Wagtail CMS. You can find out more here.

1. Install the module:

   ```
   pip install wagtail-personalisation
   ```

2. Add the module and `wagtail.contrib.modeladmin` to your `INSTALLED_APPS`:

   ```
   INSTALLED_APPS = [
       # ...
       'wagtail.contrib.modeladmin',
       'wagtail_personalisation',
       # ...
   ]
   ```

3. Update your database:

   ```
   python manage.py migrate
   ```

Continue reading: *Implementation*

CHAPTER 2

Implementation

2.1 Extending a page to be personalisable

Wagxperience offers a `PersonalisablePage` base class to extend from. This is a standard `Page` class with personalisation options added.

2.1.1 Creating a new personalisable page

Import and extend the `personalisation.models.PersonalisablePage` class to create a personalisable page.

A very simple example for a personalisable homepage:

```
from wagtail_personalisation.models import PersonalisablePage

class HomePage(PersonalisablePage):
    subtitle = models.CharField(max_length=255)
    body = RichTextField(blank=True, default='')

    content_panels = PersonalisablePage.content_panels + [
        FieldPanel('subtitle'),
        FieldPanel('body'),
    ]
```

It's just as simple as extending a standard `Page` class.

2.1.2 Migrating an existing page to be personalisable

2.2 Creating custom rules

Rules consist of two important elements, the model's fields and the `test_user` function.

A very simple example of a rule would look something like this:

```
from django.db import models
from wagtail.admin.edit_handlers import FieldPanel
from personalisation import AbstractBaseRule

class MyNewRule(AbstractBaseRule):
    field = models.BooleanField(default=False)

    panels = [
        FieldPanel('field'),
    ]

    def __init__(self, *args, **kwargs):
        super(MyNewRule, self).__init__(*args, **kwargs)

    def test_user(self, request):
        return self.field
```

As you can see, the only real requirement is the `test_user` function that will either return `True` or `False` based on the model's fields and optionally the request object.

Below is the "Time rule" model included with the module, which offers more complex functionality:

```
@python_2_unicode_compatible
class TimeRule(AbstractBaseRule):
    """Time rule to segment users based on a start and end time"""
    start_time = models.TimeField(_("Starting time"))
    end_time = models.TimeField(_("Ending time"))

    panels = [
        FieldRowPanel([
            FieldPanel('start_time'),
            FieldPanel('end_time'),
        ]),
    ]

    def __init__(self, *args, **kwargs):
        super(TimeRule, self).__init__(*args, **kwargs)

    def test_user(self, request=None):
        current_time = datetime.now().time()
        starting_time = self.start_time
        ending_time = self.end_time

        return starting_time <= current_time <= ending_time

    def __str__(self):
        return 'Time Rule'
```

Continue reading: *Usage guide*

CHAPTER 3

Usage guide

3.1 Creating a segment

As soon as the installation is completed and configured, the module will be visible in the Wagtail administrative area.

To create a segment, go to the “Segments” page and click on “Add a new segment”.

On this page you will be presented with a form. Follow these steps to create a new segment:

1. Enter a name for your segment.
2. (Optional) Select whether to match any or all defined rules.

   `match any` will result in a segment that is applied as soon as one of your rules matches the visitor. When `match all` is selected, all rules must match before the segment is applied.
3. (Optional) Set the segment persistence.

   When persistence is enabled, your segment will stick to the visitor once applied, even if the rules no longer match on the next visit.
4. Define your segment rules.

   Wagxperience comes with a basic set of *Included rules* that allow you to get started quickly. The rules you define will be evaluated once a visitor makes a request to your application.
5. Save your segment.

   Click “save” to store your segment. It will be enabled by default, unless otherwise defined.

3.2 Creating personalized content

Once you’ve created a segment you can start serving these visitors with personalised content. To do this, you can go one of two directions.

1. Create a copy of a page for your segment.

2. Create StreamField blocks only visible to your segment.
3. Create a template block only visible to your segment.

3.2.1 Method 1: Create a copy

To create a copy from a page for a specific Segment (which you can change to your liking after copying it) simply go to the Explorer section and find the page you'd wish to personalize.

You'll notice a new "Variants" dropdown button has appeared. Click the button and select the segment you'd like to create personalized content for.

Once you've selected the segment, a copy of the page will be created with a title that includes the segment. Don't worry, your visitors won't be able to see this title.

You can change everything on this page you'd like. Visitors that are included in your segment, will automatically see the new page you've created for them.

3.2.2 Method 2: Create a StreamField block

3.2.3 Method 3: Create a template block

You can add a template block that only shows its contents to users of a specific segment. This is done using the "segment" block.

When editing templates make sure to load the `wagtail_personalisation_tags` tags library in the template:

```
{% load wagtail_personalisation_tags %}
```

After that you can add a template block with the name of the segment you want the content to show up for:

```
{% segment name="My Segment" %}
    <p>Only users within "My Segment" see this!</p>
{% endsegment %}
```

The template block currently only supports one segment at a time. If you want to target multiple segments you will have to make multiple blocks with the same content.

CHAPTER 4

Included rules

4.1 Time rule

The time rule allows you to segment visitors based on the time of their visit. Define a time frame in which visitors are matched to this segment.

Option	Description
Start time	The start time of your time frame.
End time	The end time of your time frame.

```
wagtail_personalisation.rules.TimeRule
```

4.2 Day rule

The day rule allows you to segment visitors based on the day of their visit. Select one or multiple days on which you'd like your segment to be applied.

Option	Description
Monday	Matches when the visitors visits on a monday.
Tuesday	Matches when the visitors visits on a tuesday.
Wednesday	Matches when the visitors visits on a wednesday.
Thursday	Matches when the visitors visits on a thursday.
Friday	Matches when the visitors visits on a friday.
Saturday	Matches when the visitors visits on a saturday.
Sunday	Matches when the visitors visits on a sunday.

```
wagtail_personalisation.rules.DayRule
```

4.3 Referral rule

The referral rule allows you to match visitors based on the website they were referred from. For example:

```
example\.com|secondexample\.com|.*subdomain\.com
```

Option	Description
Regex string	The regex string to match the referral header to.

`wagtail_personalisation.rules.ReferralRule`

4.4 Visit count rule

The visit count rule allows you to segment a visitor based on the amount of visits per page. Use the operator to to set a maximum, minimum or equal amount of visits.

Option	Description
Page	The page on which visits will be counted.
Count	The amount of visits to match.
Operator	Whether to match for more than, less than or equal to the specified visit count.

`wagtail_personalisation.rules.VisitCountRule`

4.5 Query rule

The query rule allows you to match a visitor based on the query included in the url. It let's you define both the parameter and the value. It will look something like this:

```
example.com/?campaign=ourbestoffer
```

Option	Description
Parameter	The first part of the query ('campaign').
Value	The second part of the query ('ourbestoffer').

`wagtail_personalisation.rules.QueryRule`

4.6 Device rule

The device rule allows you to match visitors by the type of device they are using. You can select any combination you want.

Option	Description
Mobile phone	Matches when the visitor uses a mobile phone.
Tablet	Matches when the visitor uses a tablet.
Desktop	Matches when the visitor uses a desktop.

`wagtail_personalisation.rules.DeviceRule`

4.7 User is logged in rule

The user is logged in rule allows you to match visitors that are authenticated and logged in to your app.

Option	Description
Is logged in	Whether the user is logged in or logged out.

`wagtail_personalisation.rules.UserIsLoggedInRule`

CHAPTER 5

Indices and tables

- genindex
- modindex
- search