

Making Moves



Reindert-Jan Ekker

@rjekker <http://nl.linkedin.com/in/rjekker>



Summary



Making moves

- Game detail page
- Absolute URL
- block.super
- Template includes
- Custom validation
- Fat Models

Class-based Views and Generic Views



Demo



The Game Detail View

- `get_absolute_url()`
- `reverse()`
- `{{ block.super }}`



Demo



Making a Move

- A ModelForm for moves
- Custom Validation
- Overriding Model save()
- Template Includes

```
from django.urls import reverse
def get_absolute_url(self):
    return reverse('gameplay_detail', args=[self.id])
```

get_absolute_url() and reverse()

get_absolute_url() provides canonical URL for a Model instance

reverse() gets the URL for a specific View

Works like {% url %} template tag



Template: methods

You can call methods on objects from a template:

```
<a href="{{ game.get_absolute_url }}">
```

But you can NOT pass arguments to a method

This does not work:

```
{% if game.is_users_move(user) %}
```

Write this kind of logic in the Model or View



```
from django.core.validators import
    MaxValueValidator, MinValueValidator

# In Model class

y = models.IntegerField(
    validators=[MinValueValidator(0),
               MaxValueValidator(BOARD_SIZE-1)]
```

Model Validation

Django provides a set of validators -- See <https://goo.gl/TSLng6>

You can also write your own validator (it's easy)

For multiple fields: `ModelForm.clean()` -- <http://goo.gl/qbS0ZG>



Classes, not functions

Allow inheritance

<http://goo.gl/RE8ric>

Class-based Views



Demo



Generic Views

- Listing all games with ListView
- User signup with CreateView



Generic Views

Display

TemplateView

DetailView

ListView

Editing

CreateView

UpdateView

DeleteView

<http://goo.gl/h2ZC3k>



Summary



Making moves

- Game detail view
- Absolute URL and `reverse()`
- `block.super`
- Template includes
- Custom validation
- Fat models, skinny views

Class-based Views and Generic Views

