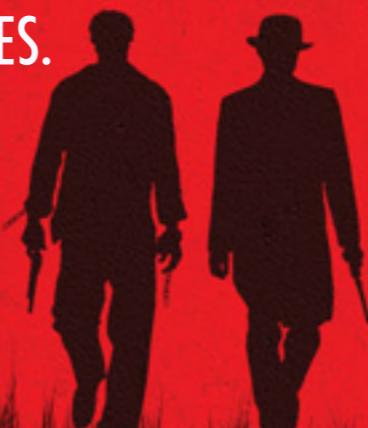


# DJANGO UNCHAINED



THE WEB FRAMEWORK FOR PERFECTIONISTS WITH DEADLINES.

NAHIM NASSER



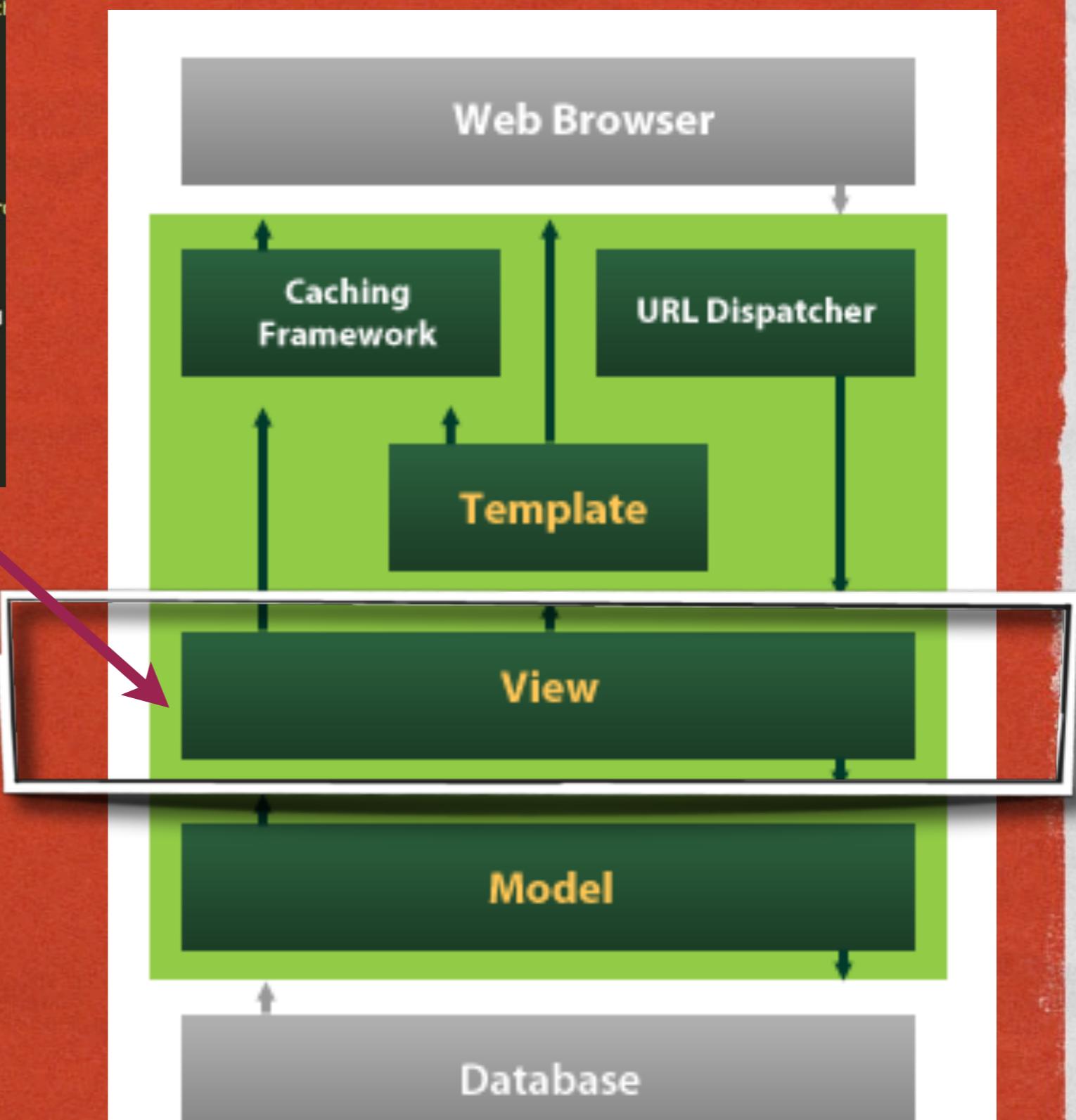
SHAH WARRAICH

EPISODE 4:  
VIEWS IN DJANGO  
&  
BUILDING AN API

# DJANGO VIEWS

# REMEMBER THIS?

```
@login_required
def end_game(request,*args,**kwargs):
    """
    A POST request that should provide the necessary parameters for the
    game_id - game id that is to be ended
    """
    user = request.user
    if request.method == 'POST':
        game_id = int(request.REQUEST.get('game_id',-1))
        response = {'success':False,'message':''}
        """
        Validate parameters, make sure the user can end the gameid provided
        """
        try:
            game = Game.objects.get(id=game_id)
        except Game.DoesNotExist:
            response['message'] = 'Game with id %s not found' % game_id
            return render_json(response,'NotFound')
        if(game.ended):
            response['message'] = 'Game has ended already.'
            return render_json(response,'BadRequest')
        if game.player2 != user:
            response['message'] = 'User %s is not player2' % user
```



# THE GLAMOROUS LIFE OF A VIEW

TAKE A WEB REQUEST,  
RETURN A WEB RESPONSE

# GOOD PRACTICES WHEN WRITING VIEWS

**RETURN CORRECT  
HTTP STATUS CODES**

**VALIDATE YOUR  
INPUT!  
(DJANGO FORMS!)**

**USE GENERIC VIEWS  
WHENEVER YOU CAN**

SO YOU WANT  
TO BUILD AN  
API?

THE CURRENT TREND

# RESTFUL API'S

# WHAT IS REST?



# REPRESENTATIONAL STATE TRANSFER

# KEY CONSTRAINTS OF REST:

- CLIENT-SERVER
- STATELESS
- CACHEABLE
- LAYERED SYSTEM
- CODE ON DEMAND
- UNIFORM INTERFACE

# HTTP METHODS

RESTful Web Service HTTP methods

Resource	GET	PUT	POST	DELETE
<b>Collection URI, such as <code>http://example.com/resources/</code></b>	List the URIs and perhaps other details of the collection's members.	Replace the entire collection with another collection.	Create a new entry in the collection. The new entry's URL is assigned automatically and is usually returned by the operation.	Delete the entire collection.
<b>Element URI, such as <code>http://example.com/resources/item17</code></b>	Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	Replace the addressed member of the collection, or if it doesn't exist, create it.	Treat the addressed member as a collection in its own right and create a new entry in it.	Delete the addressed member of the collection.

# DJANGO REST FRAMEWORK

CLEAN, SIMPLE, &  
RESTFUL

VIEWS

# DRF-View Implementation

RESTful Web Service HTTP methods				
Resource	GET	PUT	POST	DELETE
<p><b>Collection URI, such as</b> <code>http://example.com/resources/</code></p> <p><b>ListOrCreateModelView</b></p>	<p><b>List</b> the URIs and perhaps other details of the collection's members.</p>	<p>Replace the entire collection with another collection.</p>	<p><b>Create</b> a new entry in the collection. The new entry's URL is assigned automatically and is usually returned by the operation.</p>	<p>Delete the entire collection.</p>
<p><b>Element URI, such as</b> <code>http://example.com/resources/item17</code></p> <p><b>InstanceModelView</b></p>	<p>Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.</p>	<p>Replace the addressed member of the collection, or if it doesn't exist, <b>create</b> it.</p>	<p>Treat the addressed member as a collection in its own right and <b>create</b> a new entry in it.</p>	<p>Delete the addressed member of the collection.</p>

# RESOURCES

## SERIALIZING OBJECTS

A **Resource** determines how a **python object** maps to some serializable data. Objects that a resource can act on include plain Python object instances, **Django Models**, and **Django QuerySets**.

MANY MORE  
MODULES AND  
PLUGGABLE MIXINS



# SWEET HOOKS

# WALKING THROUGH THE API

<https://github.com/tomchristie/django-rest-framework/blob/master/djangorestframework/views.py>