
WTV Documentation

Release 1.3

Thomas Purchas, Josh MacNamee

January 13, 2012

CONTENTS

1	Booking System	3
1.1	The Plan	3
1.2	Models	7
2	Indices and tables	9
	Python Module Index	11

Hopeful this will start to document exactly how this site works :)

Contents:

BOOKING SYSTEM

The WTV booking system is currently a work in progress, you can find a plan and current progress here.

Note that at the moment there really isnt much of a plan, thats the first thing to the todo list :)

1.1 The Plan

The booking system needs to be able to do several things:

- Manage kit availability
- Allow kit availability to be checked
- Allow people to book kit out in the future
- Get booking approval of relevant people
- Log kit bookings and approvals

1.1.1 Usage Scenarios

These scenarios outline what I think are the most common usage scenarios, they will help define exactly what need to be achieved.

Usage 1

A person needs to book out a camera for a person project. To do this they log into the WTV website and enter the booking system.

Once in the booking system they find the camera that they want to book out using the helpful kit search tool.

After finding the camera in question the then bring up its availability which is displayed in an intuitive calendar type interface.

Now that the user know the availability of the camera they select a period of time when the camera is not booked out and create a booking. The booking page automatically pulls in all the information that the user has provided, this includes things like:

- The item to be booked out, it is resolved to an actual item - in this case the camera.
- The period that the user wishes to book the item out for.

The user then provides a reason as to why they wish to book the item out for, providing a brief description of their project.

At this point a booking object has been added to the database and the end-user's interaction with the booking system end for now. An email is sent to an appropriate administrator to review the booking request and approve/reject it.

The administrator receives the aforementioned email and reads it. It happens to contain a handy link that takes the admin straight to the booking review page to approve it.

Note: Maybe the email could contain all of the booking details, along with links to directly approve or reject the booking straight from the email.

Usage 2

A group wish to book out several pieces of kit at the same time for a large project. To start they visit the booking system on the WTV website.

They select all of the items of kit that they need for there project, on the next page the availability of all of the items is displayed in a grid display that makes it very easy to work out when all of the items will be available.

They notice that all of the items are available when they need them and for the amount of time that they need them for, except for a single camera which is only available for part of the time that they require it.

The booking tool allows them to book out all of the available kit at the same time, with the camera booked out only for the time that it is available.

All of the items including the camera appear under a single booking which makes it easy to view and approve the booking. The camera is stored under a sub-booking of the larger booking so that it is only booked out when it is available.

Usage 3

After booking out and using several items of kit the booking period starts to come to a close.

The person who booked out the kit is sent an email 2 hours before they have to return the kit warning that there time is almost up.

This email is also repeated at 1h and 30mins and 10mins before the end of the booking period, or until the kit is marked as being returned and in the store room.

If the items aren't returned by the end of the booking period the persion is emailed that they need to return the kit pronto. Also and administrator is email to warn them that kit has not been returned, and to tell them who needs to return it.

Note: The email could contain contact details for the person with the kit so that the admin can contact them personally.

If the kit is not returned before its next booking the next person to take the kit will be sent and email to warn them that they may not be able to get the kit they need. If

Note: The booking system could reallocate available kit for the next person.

Usage 4

A person wishes to book out several cables of the same type for use in a project. After logging into the booking system they select the type of cable they want from the available cables.

Once they have selected the cable they want they then enter the number they want. The booking system then generates a booking and automatically allocates the correct number of cables, from those that are available, to the user.

1.1.2 Design

To this end we will use three models to achieve the results above.

1.1.3 Models

Booking

The booking object will represent a booking made by a user. It is responsible for storing all the information related to a booking including things like:

1. User who made the booking
2. The user who approved the booking
3. The booking start time and date
4. The booking end time and date
5. The items that are to be booked
6. The reason why the user wants to book out the items.

KitType

This represents a single type of kit. It should contain all of the details relating to a single type/specification of kit.

Each KitType should be unique.

Kit

Represents a single physical object. If you have 50 cables, then each one has its own Kit object representing it.

Each Kit is linked to a KitType which provides most of the details. The main thing that a Kit contains is a serial number that makes it possible to work out exactly what physical item is being represented.

1.1.4 Views

There needs to be several views for a complete booking work flow. These break down roughly like this:

- Check kit availability
- Check multiple kit availability
- Create a new booking
- Check booking status (approved or not)
- Approve or disapprove a booking

- See all bookings
- See bookings based on a filter
- Bookings overview for a user


Kit Availability

This view needs to show the availability of a kit type over a period of time. It should show how many of that type are available to be booked out. This then needs to be replicated to show multiple kit types on the same page.

Idea

This could be shown using a calendar type view with available blocks coloured in, the colour or shade could show how many items are available in a single block. A tool-tip could then provide a precise figure.

Below is a quick mock up.



	Day 1	Day 2	Day 3	Day 4	Day 5
Kit 1	Dark Blue			Light Blue	
Kit 2	Dark Blue			Medium Blue	
Kit 3					
Kit 4					
Kit 5	Medium Blue				
Kit 6					
Kit 7	Light Blue			Very Light Blue	
Kit 8					

Create Booking

This view needs allow the user to select the items that they wish to book out. If possible the items should be derived from previous steps.

Workflow

Open the WTV site —> Login —> Open the booking system —> Select period that kit is required —> Select the kit —> Review possible time and adjust —> Create booking.

1.2 Models

`class booking.Models.Booking(models.Model)`

This object represents your basic booking object, it contains data on the start and end of the a booking, rejection or approval and other key booking information.

user

The user that created/requested the booking.

parent

The parent booking, allows booking to be grouped. This is useful if single pieces of kit are booked out for less time than other pieces of kit.

This means that infant bookings will not cover the same time period as the parent but instead only a small sub period of time.

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Parent booking	#####	#####	#####	#####	#####	
Infant			#####	#####		

: Booking in effect

Note that the infant only effects days that the parent already covers.

start_date

The date and time that the booking will start.

end_date

The date and time that the booking will end.

booking_reason

A description by the user making the booking describing why then need the kit and what they will use it for.

approved

Has the booking been approved. This is set once the booking has been reviewed by a WTV admin.

rejected

Has the booking been rejected by and admin, in which case they should have provided a reason that goes into `rejection_reason`.

approved_rejected_by

A foreign key ref pointing to the person that approved the booking.

rejection_reason

Why has the booking been rejected, this is to be shown to the `user` so that they can take appropriate action.

kits

A `ManyToManyField` pointing at all the `kit`'s that have been booked out under this booking.

It may be necessary to look infant bookings to get a complete list, note that infant bookings will not cover the same time period as the parent. See `parent` for more info.

clean()

Validation code that checks to make sure the booking is valid, to this end it checks the following:

- All kit is available to be booked out.
- That the booking is either approved or rejected, not both.
- That there is no rejection reason if the booking is not rejected.

class `booking.Models.KitType` (*models.Model*)

Represents a single type of kit e.g. A single camera model, or cable type and length.

name

Easy to read name of kit type e.g. Cable - XLR 5m

description

Detail about the kit type. Most useful for things like cameras that have numerous specifications.

class `booking.Models.Kit` (*models.Model*)

A single piece of kit. Every item that can be booked out should be represented by a `Kit` object.

type

A foreign key pointing at a `KitType`.

serial

A unique number to identify the real world bit of kit by. This *needs* to be unique and is enforced.

in_store

Is the item currently in the store room, this is not the same as being booked out, it is possible for an item to be booked out and in store at the same time.

booked_out()

Checks to see if the kit is booked out between two dates.

Parameters

- **from_date** – The date to check from.
- **to_date** – The date to check to.
- **ignore** (*Primary Key*) – A booking to ignore, used when a `Booking` performs validation.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

b

`booking.Models`, [7](#)