OpenEyes - Event Type Generation Tool

Editors: G W Aylward
Version: 0.9:
Date issued: 17 June 2012

Target Audience

General Interest	
Heathcare managers	
Ophthalmologists	✔
Developers	✔

Amendment Record

Issue	Description	Author	Date
0.9	Draft	G W Aylward	17/06/2012

Table of Contents

Introduction

This guide gives step by step instructions on how to use the OpenEyes event type generation tool. Events are a core part of OpenEyes, and most new functions will involve the creation of at least one event type.[1] Event types are implemented as Yii modules and can be created manually. However the use of the Gii tool[2] to create a new module saves a great deal of time and delivers considerably functionality 'out of the box'. This allows developers to spend more of their time delivering unique code rather than repetitive 'boilerplate'. Once the code has been auto-generated, the files are then available for customisation as required.

In this guide we will use the tool to create a simple event type for OpenEyes which will allow details of a telephone call from a patient to be recorded in the record. This may well be a real world event type since this sort of communication is often lost, a common cause of patient complaints in many hospitals!

Preparation

1. Enable Gii

Gii is a module in itself, so an entry in your configuration file needs to be made in order to allow it to run;

```
‣ edit ~/Sites/openeyesdev/openeyes/protected/config/local/
  common.php
```

Edit the configuration file so that gii is included in the array of modules;

```
	'modules' => array(
		'gii'=>array(
			'class'=>'system.gii.GiiModule',
			'password'=>'secret'
		),
	),
```

Since Gii can modify files, it is important to give access to it via a password, which should be set in the configuration file.

2. Give Gii write permissions

Since Gii will be creating files in the modules directory, check that the webserver has write permissions for the modules directory. If it does not, add them with the following commands;

```
‣ cd ~/Sites/openeyesdev/openeyes/protected
‣ chmod 775 modules
```

Design

Our new event will of a single element called 'Call Details' which consists of the following three fields;

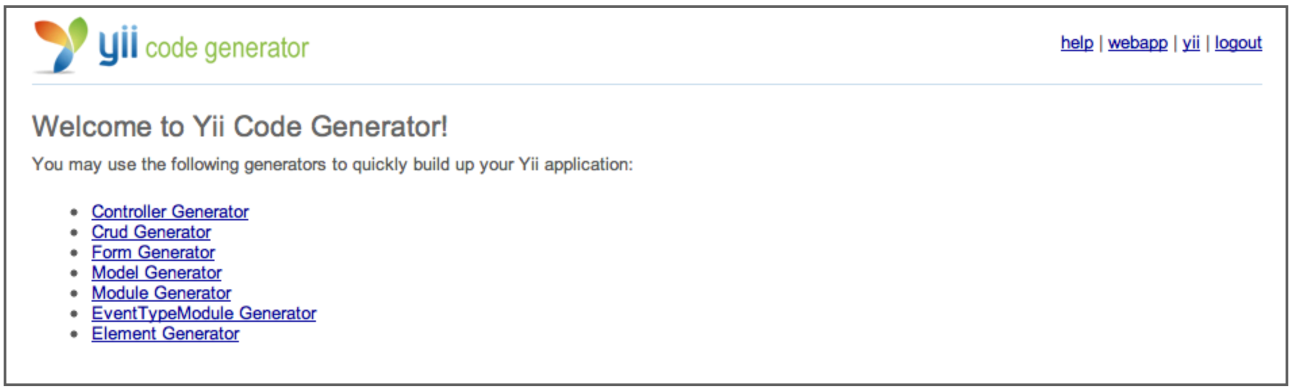
Name	Type	Description
Conversation	Text	A text box allowing entry of notes from the conversation
Priority	Drop down	A drop down indicating the urgency of the call
Action	Boolean	A check box indicating that any action from the call has or has not been taken

Run Gii

Gii can be run using the following URL assuming you are running OpenEyes on localhost. If not adjust the URL accordingly.

http://openeyes.localdomain/gii/default/login

Enter the password (defined in the configuration file) and you will be taken to the following screen;

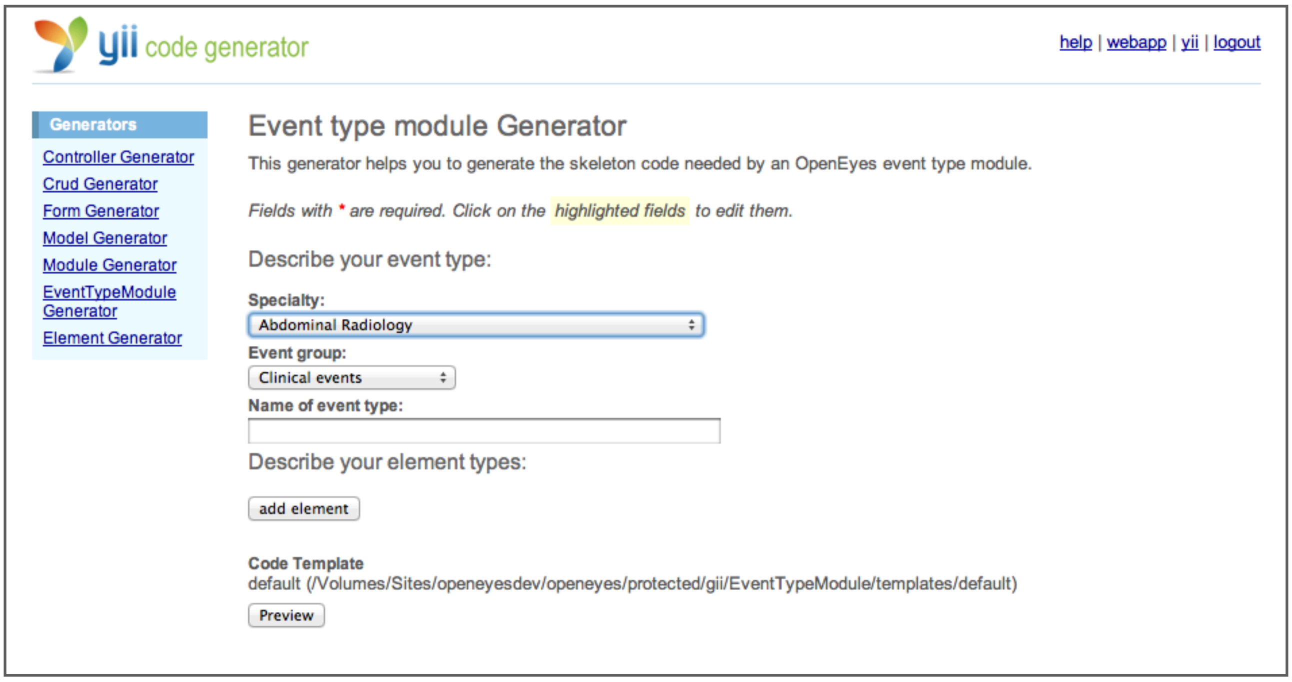


Welcome to Yii Code Generator!

You may use the following generators to quickly build up your Yii application:

- Controller Generator
- Crud Generator
- Form Generator
- Model Generator
- Module Generator
- EventTypeModule Generator
- Element Generator

Click on EventTypeModule Generator which will result in the following screen;



Event type module Generator

This generator helps you to generate the skeleton code needed by an OpenEyes event type module.

*Fields with * are required. Click on the highlighted fields to edit them.*

Describe your event type:

Specialty: Abdominal Radiology

Event group: Clinical events

Name of event type:

Describe your element types:

Code Template
default (/Volumes/Sites/openeyesdev/openeyes/protected/gii/EventTypeModule/templates/default)

Define the event

1. From the Specialty drop down choose Ophthalmology

All other medical specialties are available since it is possible that OpenEyes will be used outside of ophthalmology, and the Gii tool will take care of possible future namespace conflicts (e.g. the examination event type will be different between specialties.

2. Choose Communication event from the Event group drop down

OpenEyes supports the concept of event groups. These are displayed using a different colour for each group so that information on the event groups can be subtly conveyed.

3. Name the event

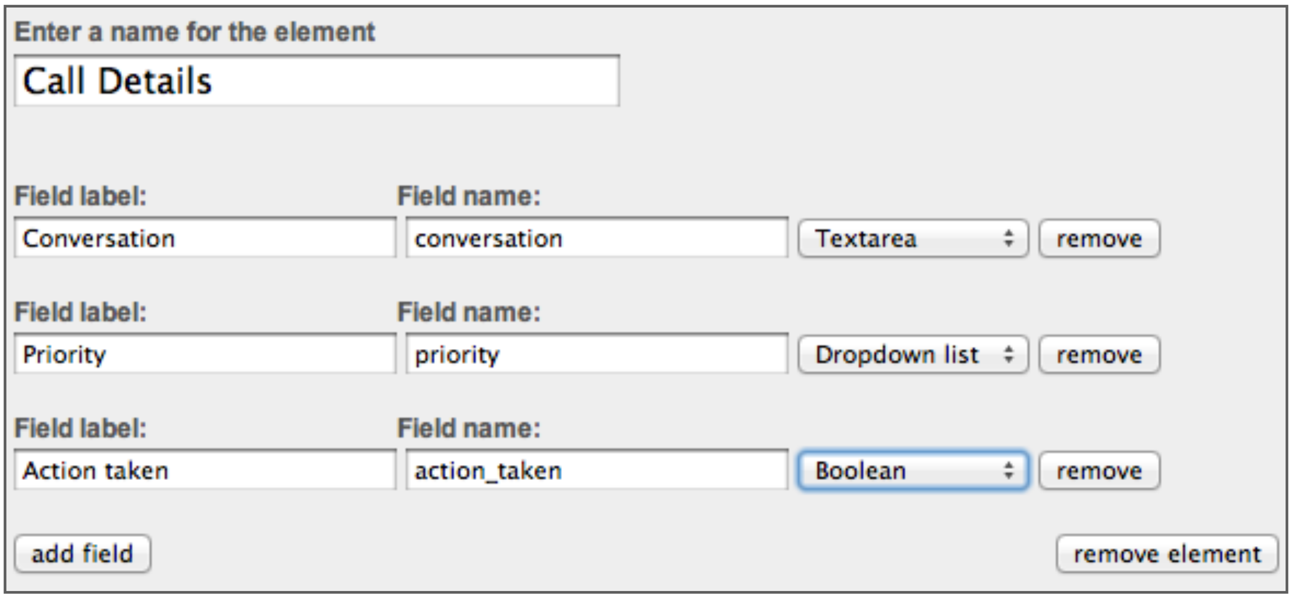
Name the event type ‘Telephone Call’.

4. Add an element

No click the add element button and name it ‘Call Details’

5. Add Fields

Using the ‘Add field’ button, add the three fields in the table in the design section above. The screen should now look like this;


Enter a name for the element
Call Details

Field label:	Field name:		
Conversation	conversation	Textarea	remove
Priority	priority	Dropdown list	remove
Action taken	action_taken	Boolean	remove

Generate the code

Click the 'Preview button' and you will be presented with a list of the files that will be generated. Give it a sense check and then press the 'Generate button'.

Configure

You now need to configure the system to make use of the new module. Instructions are displayed in the browser after the last step

1. Add the module to the module array

Edit the configuration file to ensure the new module is in the module array;

```
'modules' => array(
        'OphCoTelephoneCall',
        'gii'=>array(
                'class'=>'system.gii.GiiModule',
                'password'=>'secret'
        ),
),
```

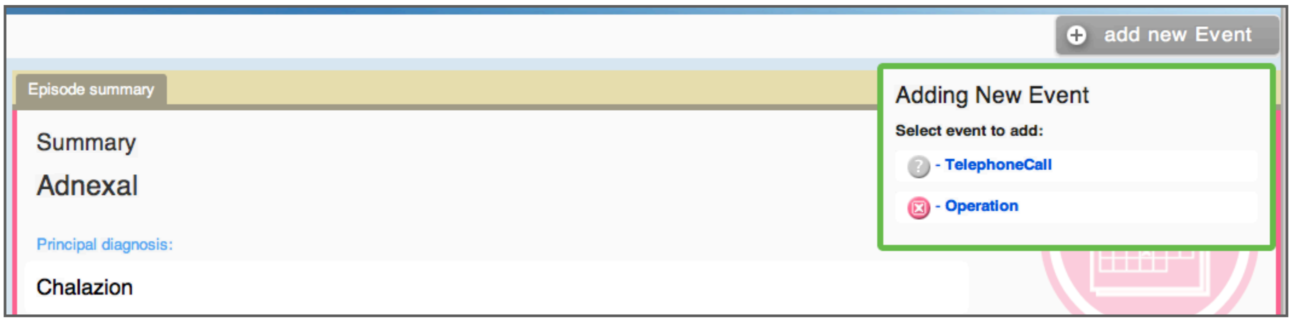
2. Add tables to the database

Use the migration script to add tables to the database;

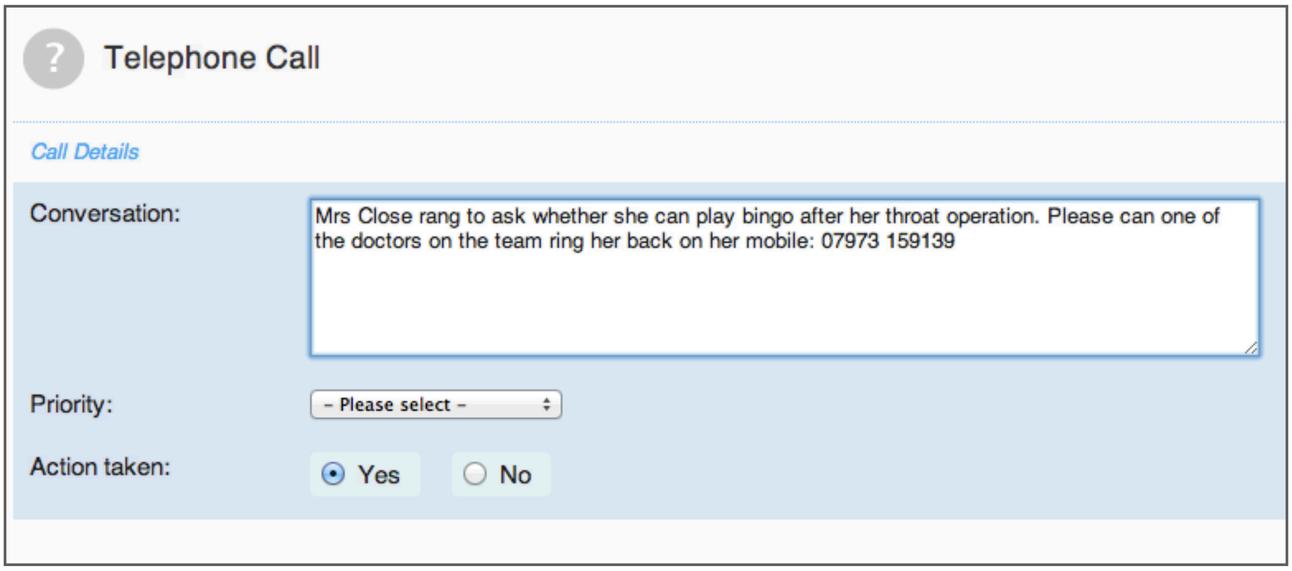
```
‣ cd ~/Sites/openeyesdev/openeyes/protected
‣ ./yiic migrate --
  migrationPath=application.modules.OphCoTelephoneCall.migrations
```

Test

Login to OpenEyes, find a patient and go into event view. Clicking ‘Add event’ will show that your new event type is now installed and ready to add;



Click on TelephoneCall to add the event to the patient and allow fields to be added;



Telephone Call

Call Details

Conversation: Mrs Close rang to ask whether she can play bingo after her throat operation. Please can one of the doctors on the team ring her back on her mobile: 07973 159139

Priority: - Please select -

Action taken: [✓ Yes] No

Removing a module

In order to remove a module you need to reverse some of the steps described above;

1. Remove the tables from the database

This is done by running the migration again with the addition of the ‘down’ keyword;

```
‣ ./yiic migrate down --
  migrationPath=application.modules.OphCoTelephoneCall.migrations
```

2. Remove the module reference

Edit the configuration file to remove the reference to the module which was added previously

3. Delete files

Delete the module and all its files;

```
‣ cd ~/Sites/openeyesdev/openeyes/protected/modules
‣ rm -r OphCoTelephoneCall
```

References

1. OpenEyes Clinical Events - link
2. Automatic Code Generation using Gii - link