Управление версиями

Если вы строите свой API без версии, это ужасно. Давайте представим, что вы выталкиваете критическое изменение-в основном любое изменение, которое противоречит тому, что разработчики клиентов запланировали, например, переименование или удаление параметра или изменение формата ответа - вы рискуете сбить многие, если не все, системы Ваших клиентов, что приводит к сердитым вызовам поддержки или, что еще хуже, массовому оттоку. Вот почему вы должны поддерживать свою версию API. В Yii2 управление версиями может быть легко выполнено с помощью модулей, поэтому версии будут представлены как изолированный блок кода.

Подготовка

Повторите все шаги, описанные в разделе создание рецепта REST-сервера подготовка и как это сделать...

Как это сделать...

1. Создайте следующую структуру в папке приложения. В общей сложности, вы должны создать папку @app/modules с папками v1 и v2 внутри него. В папке каждого модуля необходимо создать папки контроллеров и модели:

app/

modules/

v1/

controllers/

FilmController.php  
Module.php

v2/

controllers/

FilmController.php  
Module.php

1. Добавьте модули импорта в @app/config/web.php:

'modules' => [

'v1' => [

'class' => 'app\modules\v1\Module',

],

' v2' => [

'class' => 'app\modules\v2\Module'

]

],

1. Создайте @app/modules/v1/controllers/FilmController.php и @app/modules/v2/controllers/FilmController.php со следующим кодом:

<?php

namespace app\modules\v1\controllers;  
use yii\rest\ActiveController;

class FilmController extends ActiveController  
{

public $modelClass = 'app\models\Film';

}

<?php

namespace app\modules\v1\controllers;  
use yii\rest\ActiveController;

class FilmController extends ActiveController  
{

public $modelClass = 'app\models\Film';

}

<?php

namespace app\modules\v1;

class Module extends \yii\base\Module  
{

public function init()

{

parent::init();

}

}

<?php

namespace app\modules\v2;

class Module extends \yii\base\Module  
{

public function init()

{

parent::init();

}

}

Создайте @app/modules/v1/Module.php и @app/modules/v2/Module.php со следующим кодом:

Как это работает...

Каждый модуль представляет собой независимую версию нашего API.

Теперь вы сможете указать версию API двумя способами:

1. По URL API. Можно указать версии v1 или v2. В результате http://yii-book.app/v1/film вернет список фильмов для версии 1 и http://yii-book.app/v2/film сделает это для версии 2.
2. Можно также поместить номер версии через заголовки запроса HTTP. Как обычно, это можно сделать через заголовок Accept:

// as a vendor content type

Accept: application/vnd.company.myproject-v1+j son // via a parameter

Accept: application/json; version=v1

Итак, теперь у нас есть две версии нашего API, и мы можем легко изменить версию v2 без каких-либо головных болей. Наши старые клиенты продолжают работать с версией v1, а новые клиенты или те, кто хотел бы обновить будет использовать версию v2.

Кое что еще...

Для получения дополнительной информации обратитесь к:

* <http://www.yiiframework.com/doc-2.0/guide-rest-versioning.html>
  1. по русски http://yiiframework.domain-na.me/doc/guide/2.0/ru/rest-versioning
* <http://budiirawan.com/setup-restful-api-yii2/>