



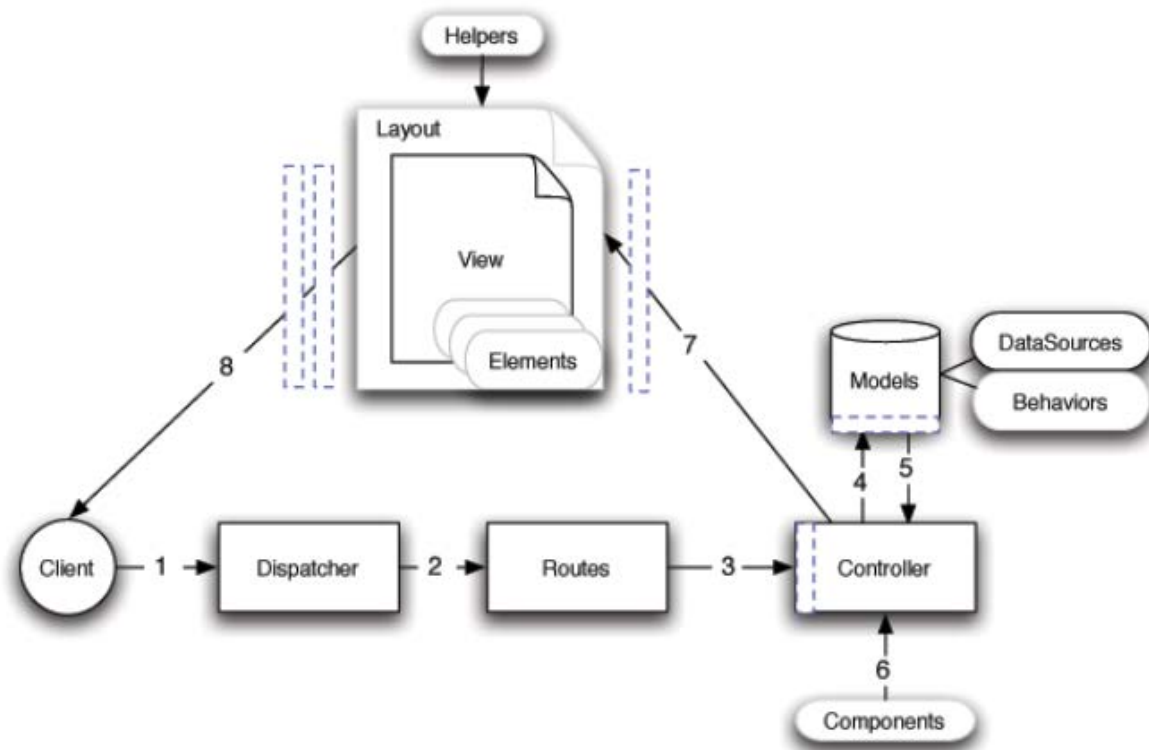
Controllers

Jill Gundersen

pluralsight 
hardcore developer training

CakePHP Request

- More Detailed Version of the CakePHP Request



Routes

- **Parse the Request URL**
 - Extracts parameters
 - e.g. controller, action, additional values passed in the URL.
 - `http://catalog/items/view/12`
- **The Request URL Parameters are then Mapped to a Controller Action**
 - A specific action in a specific controller class.
 - items controller
 - view action (function)
 - 12 value to use with the database

Controller

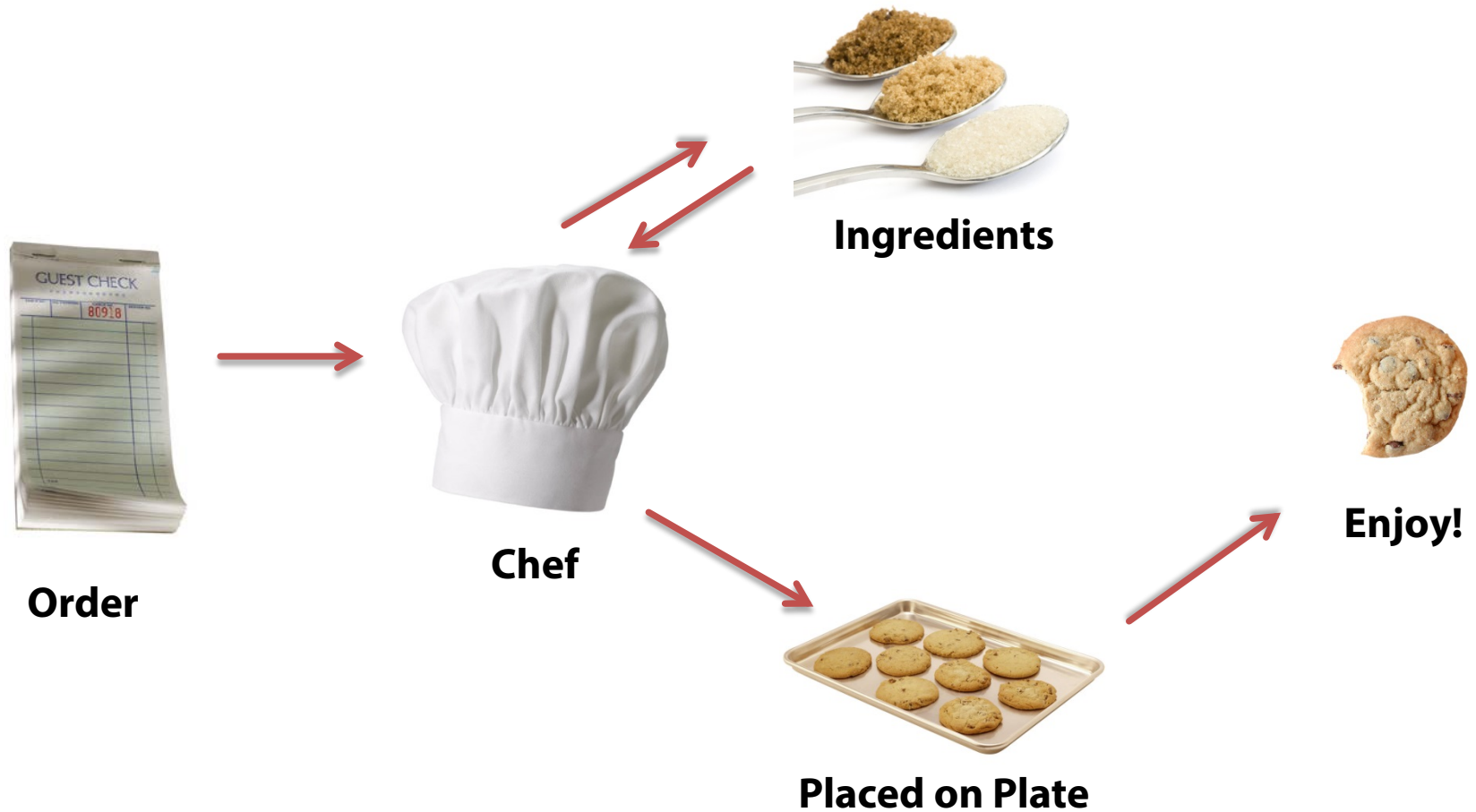
- **The Controller Interacts With the Model**

- Controller calls on the model if there is a need to gain access to the application's data.
- Model interacts with the database and returns data back to the controller.

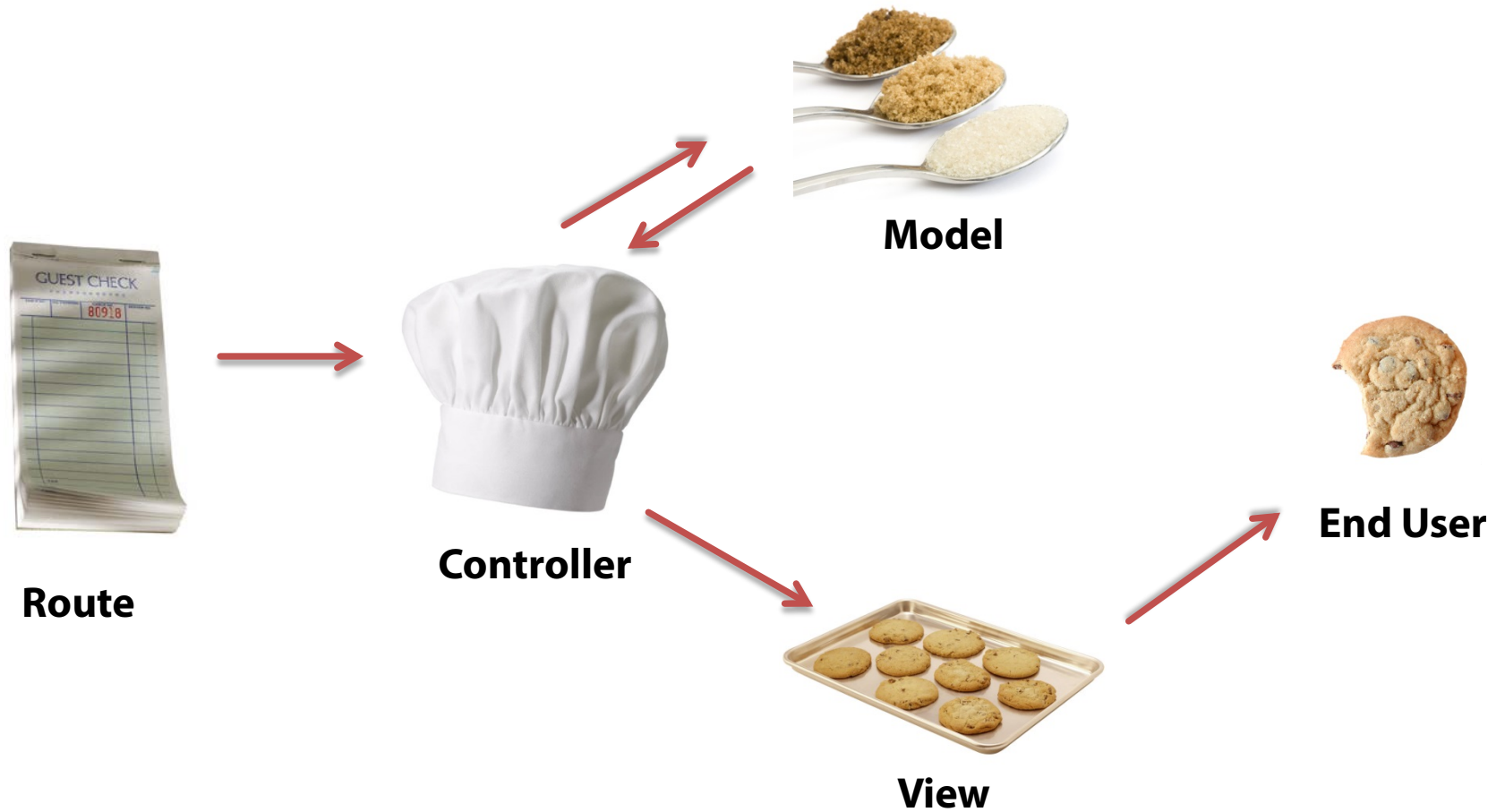
- **The Controller then Interacts With the View**

- Controller can use the data from the model and manipulate it for the benefit of the view
- After the controller has the prepared the data, the controller passes the data off to the view

A Tasty Example



A Tasty Example



What's In Store?

- **Continue Development of Our Catalog Web Site**
 - DVDs, Books, CDs, etc.
- **Add to the Controller**
 - Add Actions (functions) to the ItemsController we made in “Getting Started” module
 - Display a List all the Items
 - Create and Delete an Item
 - Update and Display a Detailed View of an Item
 - Search the Items

Setting Variables

■ Passing Variables to the View

- The set() method is used to pass variables to the view.
 - First Parameter
 - Provide a name
 - Second parameter
 - Provide the value or data
- Example
 - Controller
 - `$this->set('pizza', 'pepperoni');`
 - View
 - Their favorite pizza is `<? echo $pizza; ?>`.

Query the Database

- **Accessing the Database Model in Controller**
 - Access the Model via `$this->ModelName`
 - Example
 - `$this->Item`
 - Retrieve Data from the Model Using `find()` Method
 - Example

```
$this->Item->find('all');
```

Query the Database

■ find() Parameters

- First Parameter:
 - all / first / count / neighbors / list / threaded
- Second Parameter:
 - Array of Data
 - conditions / fields / joins / limit / offset / order / page / group / callbacks
- Example

```
$this->Item->find('all',  
    array('order' => 'title DESC'));
```

Single Query

- **Query Based on ID**
 - findById()
 - Example
 - `$this->ModelName->findById($id);`

Throw an Exception

- **CakePHP Has Many Error Exceptions**

- <http://book.cakephp.org/2.0/en/development/exceptions.html>

- **Handling Errors**

- throw new NotFoundException
 - Example
 - `throw new NotFoundException(__('Your item was not found'));`

Form Data

■ CakeRequest

- Default request object in CakePHP. Passed by reference to the various layers of an application.
- Assigned to `$this->request`
- Example
 - Holds all the data from the form (modified and unmodified)
 - `$this->request->data`

■ Check How the Data Was Sent

- `is()`
- Takes One Parameter
 - `post / get / put / delete / head / options / ajax / ssl / flash / mobile`
- Example
 - `$this->request->is('post');`

Create & Save

■ Adding a New Item Into a Database

- `create()`
- Resets the model state for saving new information. Clears out the model `$id` value
- Example
 - `$this->ModelName->create();`

■ Saving the Data

- `save()`
- Parameters
 - 1st: Array-formatted data to be saved.
 - 2nd: Allows you to side step validation (bool) – optional
 - 3rd: Supply a list of model fields to be saved. – optional
- Example
 - `$this->ModelName->save($this->request->data);`

Redirect

■ Sending a User to a Different Page

- Sometimes it is necessary
- `redirect()`
- Parameters:
 - String action (if in same controller)
 - Array controller & action
 - Relative path
- Example
 - `$this->redirect('index');`
 - `$this->redirect(array('controller' => 'Items', 'action' => 'index'));`
 - `$this->redirect('/Items/thanks');`
- More Redirect Examples
 - <http://book.cakephp.org/2.0/en/controllers.html#flow-control>

Saving An Existing Entry

- **Saving an Item**

- Use the same save method that we utilized in the add() action, but we eliminate the need for the create() method to be called before hand.
- Example
 - `$this->ModelName->save($this->request->data);`
- Override the 'id' in our model
 - Example
 - `$this->ModelName->id = $id;`

Populating the Form

- **Editing a Form**

- When you edit an item you need to populate the form with the content of the item you are editing.
- Example
 - `$item = $this->modelName->findById($id);`
 - `$this->request->data = $item;`

Set and Display Message

- **Session Component**

- CakePHP provides a messaging system with the session component

- **Adding the Session Component to Your Controller**

- `public $components = array('Session');`

- **Using the Session Component**

- Example

- `$this->Session`

- Access the Method

- `setFlash()`

- Parameter

- String message you want to display

- Example

- `$this->Session->setFlash(__('Your record was saved.'));`

Does it Exist?

■ Previous Determination

- `$data = $this->ModelName->findById($id)`
- `if(!$data)`
 - Throw Exception

■ Check to See if an Item Exists in the Database

- `exists()`
- Returns Bool Value
- Parameters
 - `$id` (optional)
- Examples
 - `$this->ModelName->exists($id);`
 - `$this->ModelName->id = $id;`
 - `$this->ModelName->exists()`

Delete Item

- **Deleting an Item in the Database**

- delete()
- Returns Bool Value
- Parameters
 - \$id (optional)
- Examples
 - `$this->ModelName->delete($id);`
 - `$this->ModelName->id = $id;`
 - `$this->ModelName->delete()`

Render Method

■ Normal Behavior

- Calls the view template of the same name as the action
 - `index()` => `index.ctp`
- Called automatically at the end of the action in the controller

■ Override

- Reuse, Plugins, Elements, etc.
- `render()`
- Parameter
 - Name of the template you would like to use
- Example
 - `$this->render('index');`
 - `$this->render('/Cakes/index');`
 - `app/View/Cakes/index`

Summary

- **A Review of the MVC Framework**
 - Better understanding of the role of the controller
- **Added Actions (Functions) to Our Items Controller**
 - Basic CRUD actions
 - Create
 - Read
 - Update
 - Delete
 - Search