

Cooking

With the CiviCRM API

Peter
McAndrew

Eileen
McNaughton

Tim
Otten

Agenda

- API Overview
- Case Study: CiviMobile
- What's New: Documentation & Validation
- Coming Soon: Native Modules

Environments

- PHP
 - Functions
 - **New:** Classes
- REST
- JavaScript
- Smarty
 - **New:** Drupal Pages
- **New:** Command Line
- **New:** Cron

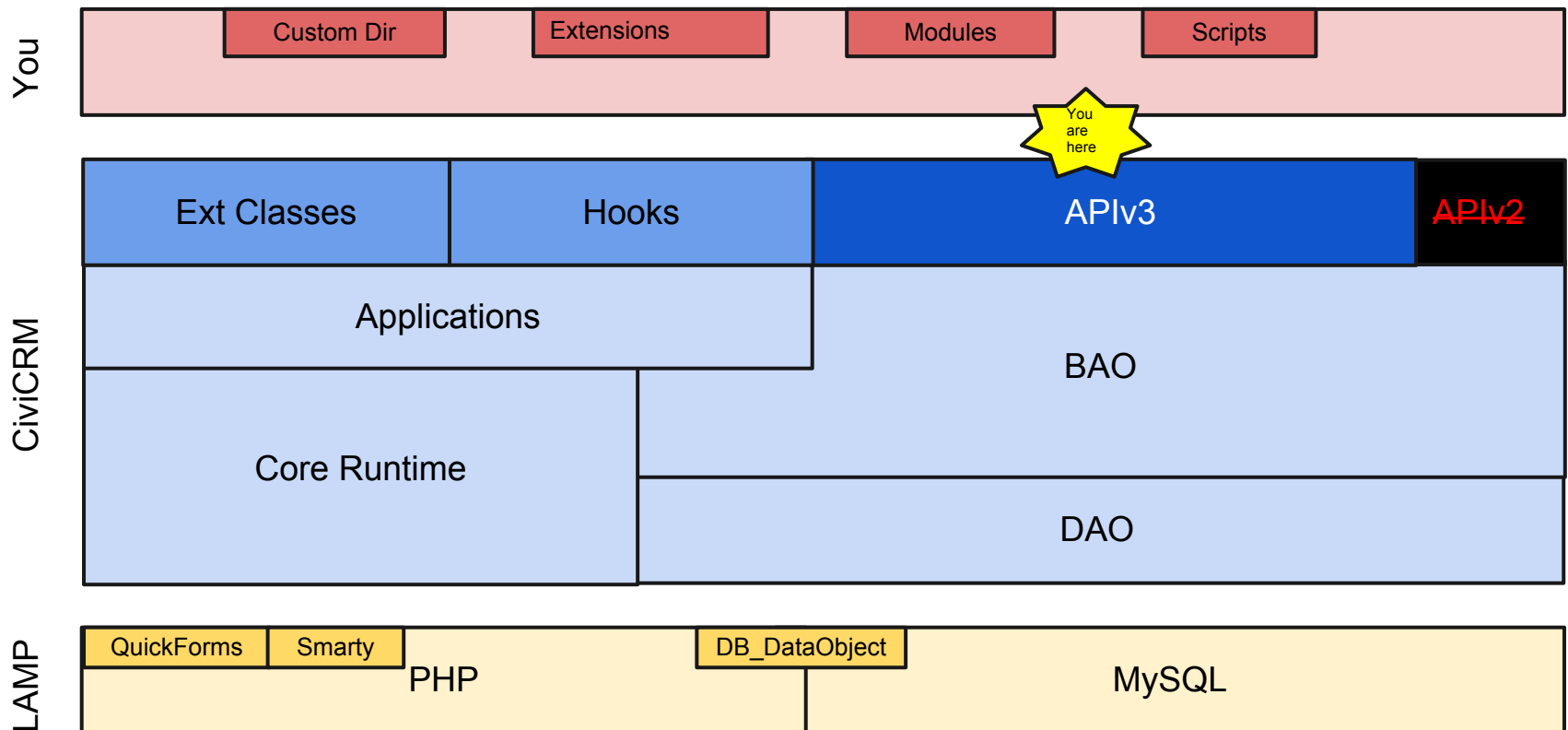
Examples

- CiviMobile
- AJAX Click-to-edit
- AJAX Calendar
- Data import
- Data synchronization
- Enriching current pages
- Recurring activities
- Data-driven pages

Examples - CiviMobile



Architecture



(split)

Documentation

PHP Doc

<http://api.civicrm.org/v3/>

API Explorer <http://drupal.demo.civicrm.org/civicrm/api/explorer#explorer>

Wiki <http://wiki.civicrm.org/confluence/display/CRMDOC41/CiviCRM+Public+APIs>

Developer Guide

<http://book.civicrm.org/developer/techniques/api>

Examples

<https://svn.watermelons.net.au/greens.org.au/trunk/sites/all/modules/civicrm/api/v3/examples>

See also: <http://bit.ly/H9a8yY>

Code Generated Documentation

Code changes. If we make the documentation part of the code they will stay in sync.

- Generated by tests
- Generated by 'getfields'

Test Generated

- For basic functions
- For specific features

```
/*
 * demonstrates use of return_is_full
 */
function event_getsingle_example() {
    $params = array(
        'id' => 1,
        'version' => 3,
        'return.is_full' => 1,
    );

    require_once 'api/api.php';
    $result = civicrm_api('event', 'getsingle', $params);

    return $result;
}
```

PHP Doc

▶ `civicrm_api3_note_create` (line 60)

Create Note

This API is used for creating a note. Required parameters : `entity_id` AND `note`

- **return:** API result array
- **access:** public
- **example:** [Create example](#)

`array civicrm_api3_note_create (array $params)`

- **array \$params:** an associative array of name/value property values of `civicrm_note`

<i>key</i>	<i>type</i>	<i>description</i>	<i>default</i>
<code>'id'</code>	<i>int</i>		
<code>'entity_table'</code>	<i>string</i>	Entity Table	<code>civicrm_contact</code>
<code>'entity_id'</code>	<i>int</i>	Required	
<code>'note'</code>	<i>string</i>	Required Note	
<code>'contact_id'</code>	<i>int</i>		
<code>'modified_date'</code>	<i>4</i>	Modified Date	<code>now</code>
<code>'subject'</code>	<i>string</i>	Subject	
<code>'privacy'</code>	<i>string</i>	Privacy	

API Explorer

The API explorer shows fields on Your site

The screenshot shows the CiviCRM 4.1 Demo Site interface. The top navigation bar includes the CiviCRM logo, the site name "CiviCRM 4.1 Demo Site", and links for "My account" and "Log out". The main content area is titled "API explorer and generator" and features a breadcrumb "Home > CiviCRM".

On the left sidebar, there is a "Create New" button, a "Recent Items" section with a user profile for "Sam Iam", and a "Navigation" section with a link to "CiviCRM".

The main content area contains the following form and text:

entity action debug sequential json

Available fields (click on it to add it to the query): **Internal Contact ID** Contact Type Contact Subtype Do Not Email Do Not Phone Do Not Mail Do Not Sms Do Not Trade **No Bulk Emails (User Opt Out)** Legal Identifier External Identifier Sort Name Display Name Nick Name Legal Name Image Url Preferred Communication Method Preferred Language Preferred Mail Format Contact Hash Api Key Source of Contact Data First Name Middle Name Last Name Individual Prefix Individual Suffix email_greeting_id Email Greeting Custom Email Greeting postal_greeting_id Postal Greeting Custom Postal Greeting addressee_id Addressee Custom Addressee Job Title Gender Birth Date Is Deceased Deceased Date Household Name primary_contact_id Organization Name Sic Code Unique ID (OpenID) Current Employer ID **Contact is in Trash** Most Important Issue Marital Status Marriage Date How long have you been a donor? Known areas of interest

You can choose an entity and an action (eg Tag Get to retrieve a list of the tags)
Or you can directly modify the url in the field above and press enter.

When you use the create method, it displays the list of existing fields for this entity.
click on the name of the fields you want to populate, fill the value(s) and press enter

The result of the ajax calls are displayed in this grey area.

Or you can just ask the API

```
$params = array(  
  'version' => 3,  
  'action' => 'create',  
);
```

```
$result = civicrm_api( 'group','getfields',$params );
```

The API implements getfields

- When the API is used it calls 'getfields' when setting defaults & enforcing requirements.
- Code & documentation is in sync

```
'is_active' => array(  
    'api.default' => 1,  
),  
'title' => array(  
    'api.required' => 1,  
),
```

Further Reading

- On getfields

<http://civicrm.org/blogs/eileen/api-civicrm-41-under-hood>