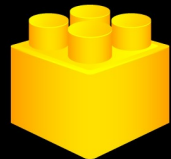


VIEWS for HACKERS

an overview of Views programming
v1.3



Karim Ratib (infojunkie@drupal)
There's a module for that!

OBJECTIVES OF THIS TALK

conceptual grasp of Views internal architecture

ability to read Views and add-ons code

ability to start coding Views add-ons

WHO AM I?

Drupal user since 2005
infojunkie (uid 48424)

Module author/maintainer since 2007

Views Bulk Operations

Views Hacks

Sheetnode

Feeds Oauth

Feeds Topics

...

Drupal freelancer since 2008

There's a module for that!

Vancouver, BC

OUTLINE

CONTEXT

problem

trivia

definition

ARCHITECTURE

overview

components

workflow

file structure

EXAMPLES

anatomy of Views add-ons

CONTEXT // problem

coding node lists is boring

coding node lists with filters is boring + 1

coding node lists with different themes is boring x 10

...

merlinofchaos chose to code all this for us!

CONTEXT // trivia

First CVS checkin: Fri Nov 25 20:33:09 2005 UTC by merlinofchaos

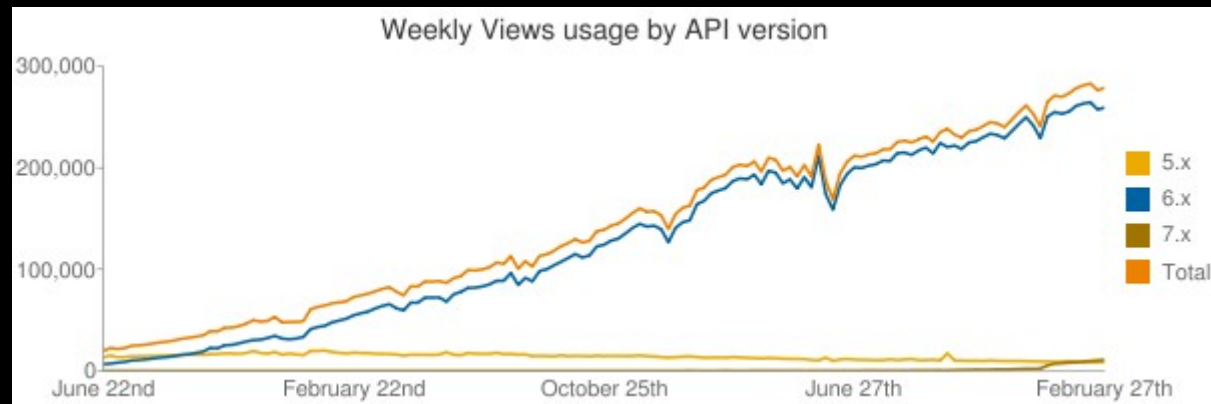
Lines of code: **95K+**

```
wc -l `find ./ -type f`
```

3rd party sub-modules: **400+**

<http://drupal.org/project/modules?filters=tid:89>

Views
282K
Drupal
397K
CCK
269K



CONTEXT // definition

Views is a
SQL query
generator
and
renderer

ARCHITECTURE // overview // 1

```
SELECT
n.title, u.name
FROM
{node} n
INNER JOIN
{users} u ON n.uid = u.uid
WHERE
n.status = 1
AND
u.uid = arg(1)
ORDER BY
n.changed DESC
```

```
hook_menu() => nodes/list
theme('table', $results)
```

ARCHITECTURE // overview // 2

```
SELECT
n.title, u.name    field
FROM
{node} n    base table = view type
INNER JOIN
{users} u ON n.uid = u.uid    relationship
WHERE
n.status = 1    filter
AND
u.uid = arg(1)    argument
ORDER BY
n.changed DESC    sort

hook_menu() => nodes/list    display (page)
theme('table', $results)    style (table)
```

ARCHITECTURE // overview // 3

The screenshot shows the configuration interface for a Drupal View named 'nodes_list'. The view is currently configured as a 'Page' display type. The interface is divided into several sections:

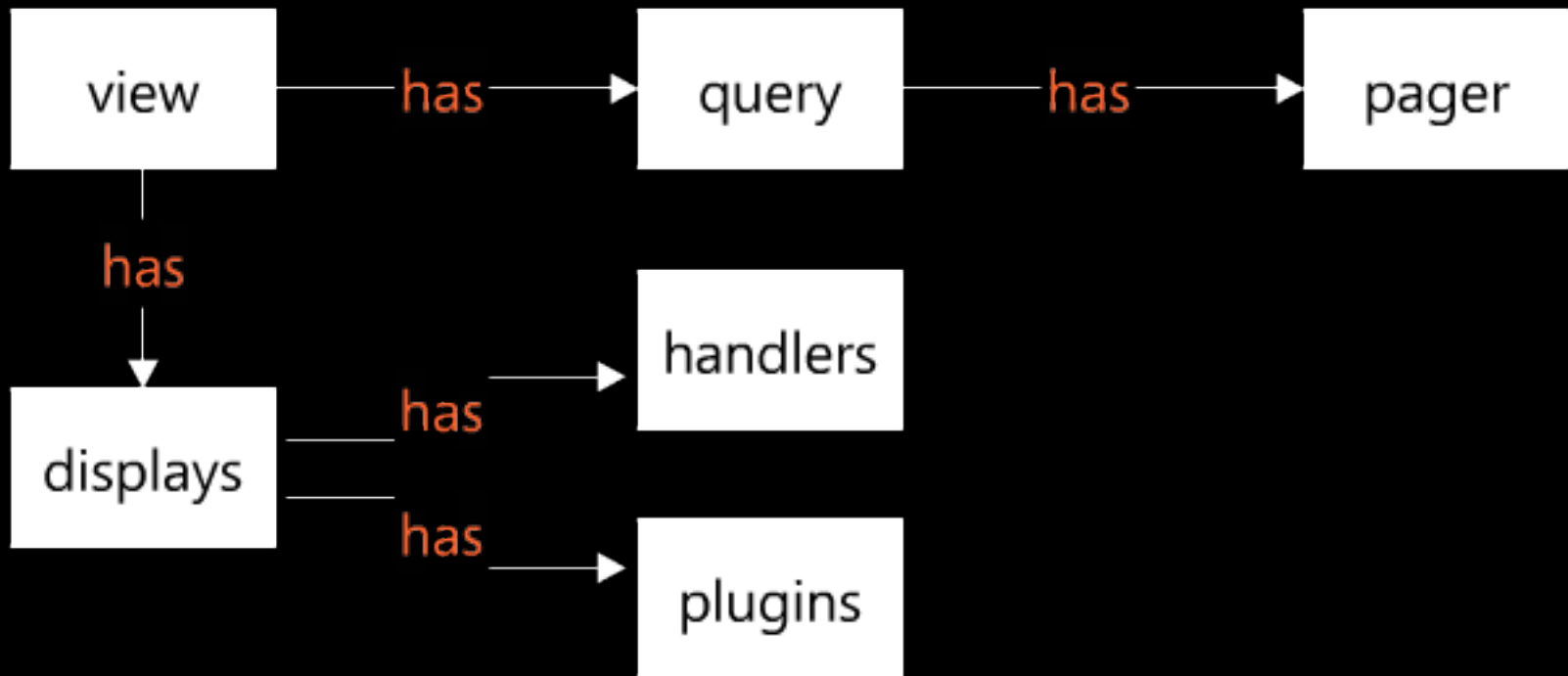
- Defaults:** Shows the current display type as 'Page' and includes an 'Add display' button.
- Page:** A sub-section for page-specific settings, including an 'Analyze' button.
- Basic settings:** A list of configuration options for the display, such as Name (Page), Title (None), Style (Table), Use AJAX (No), Use pager (No), Items to display (20), Distinct (No), Access (Unrestricted), Caching (None), Exposed form in block (No), Header (None), Footer (None), Empty text (None), CSS class (None), and Theme (Information).
- Relationships:** A section for defining relationships between fields, currently showing 'None defined'.
- Arguments:** A section for defining arguments for the display, currently showing 'User: Uid'.
- Fields:** A section for defining fields to display, currently showing 'Node: Title Title' and 'User: Name Name'.
- Sort criteria:** A section for defining sort criteria, currently showing 'Node: Updated date desc'.
- Filters:** A section for defining filters, currently showing 'Node: Published Yes'.

At the top right, there are buttons for 'Export', 'Clone', and 'View "Page"'. Below the 'Page' section, there are buttons for 'Clone display' and 'Remove display'.

same concepts are found on the Views UI view type, display, style, fields, filters, etc.

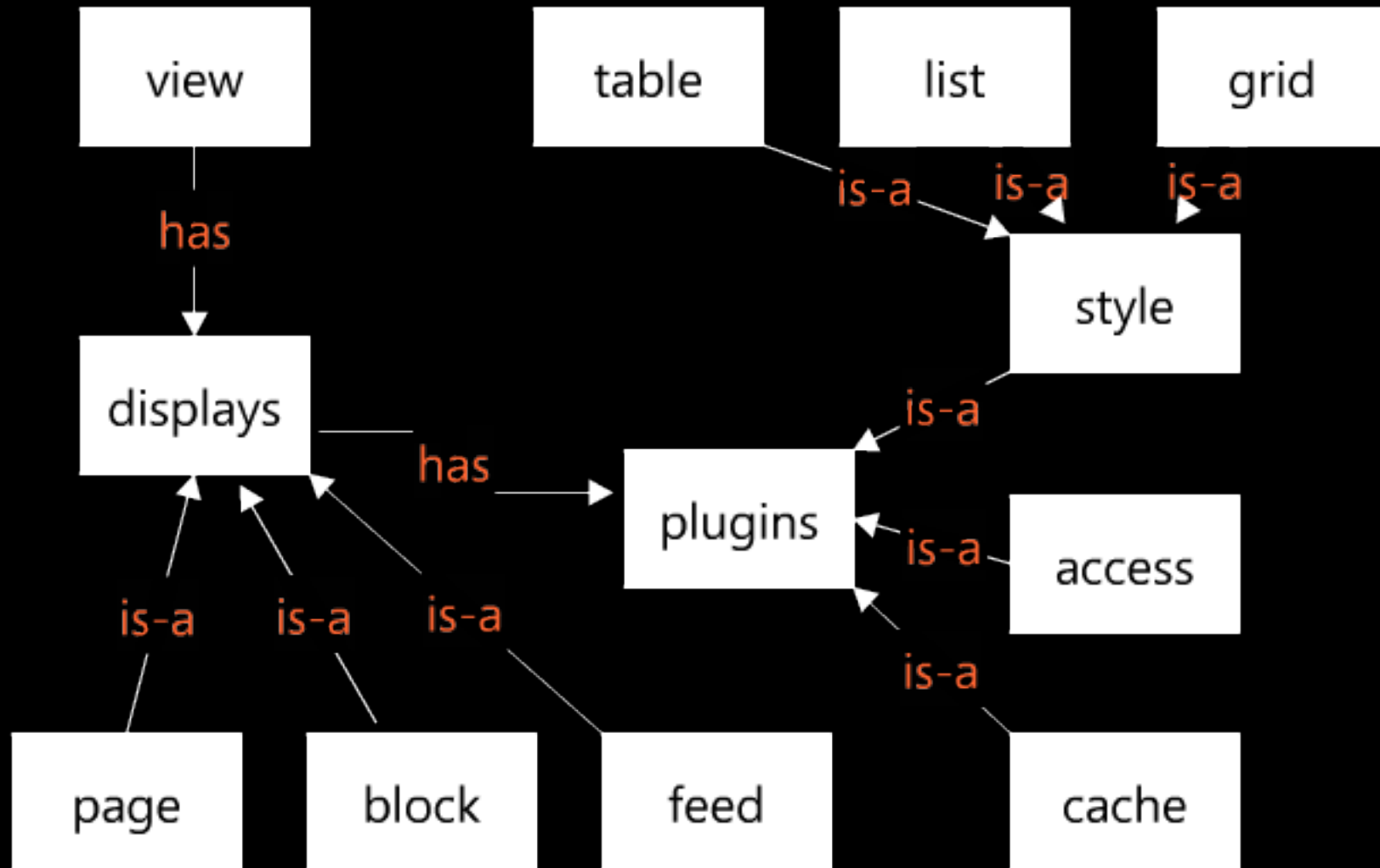
ARCHITECTURE // components // 1

containment relationships



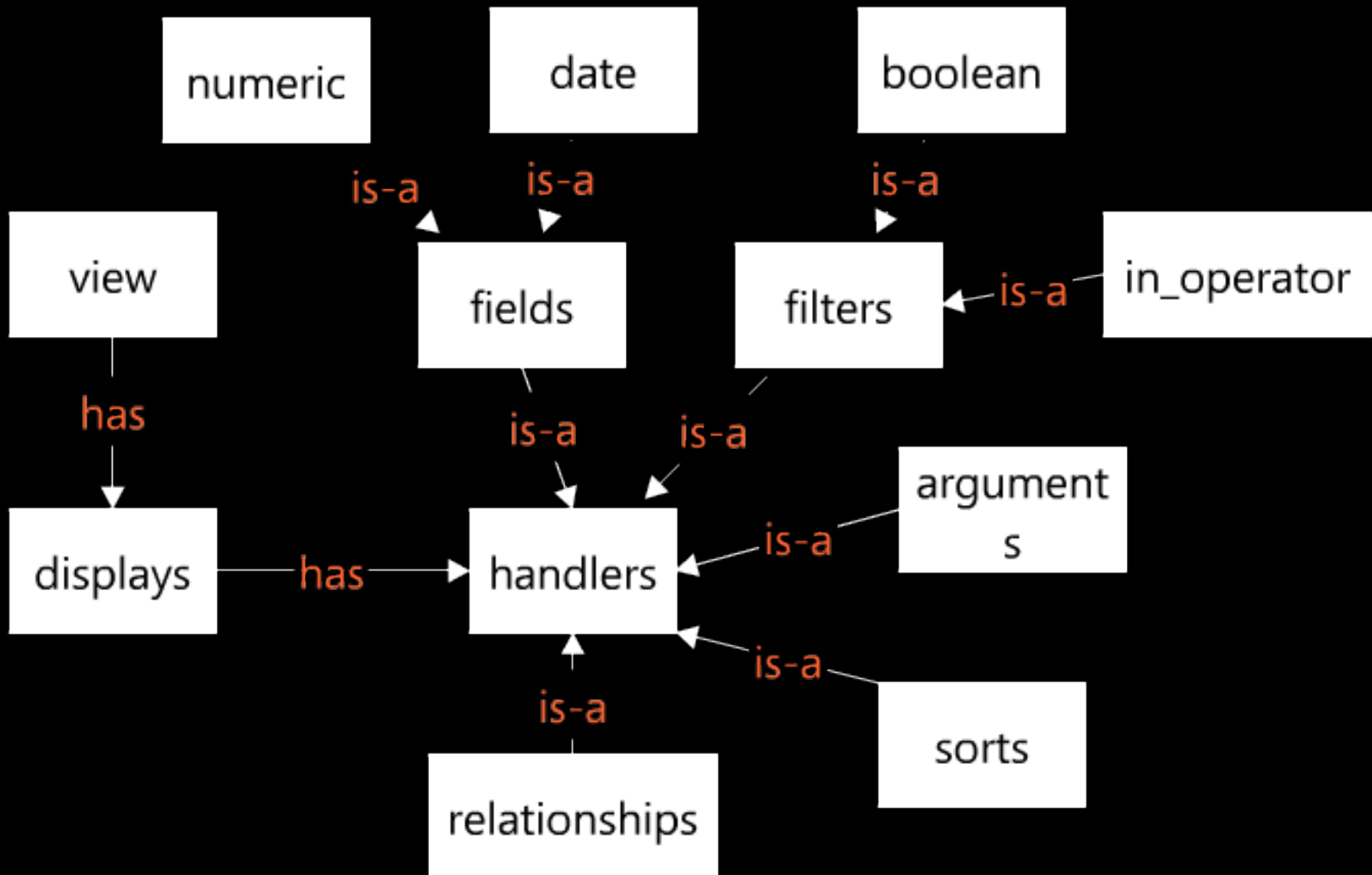
ARCHITECTURE // components // 2

plugin type relationships



ARCHITECTURE // components // 3

handler type relationships



ARCHITECTURE // workflow

BUILD

init components, get arguments, generate SQL

view::build()

hook_views_pre_build(), post_build()
hook_views_query_alter()

EXECUTE

execute SQL, store results

view::execute()

hook_views_pre_execute(), post_execute()

RENDER

execute theme functions

view::render()

hook_views_pre_render(), post_render()

ARCHITECTURE // file structure

views/ Drupal façade

- ./includes** core: view, query, base classes
- ./handlers** SQL add-ons: fields, filters, arguments, relationships, sort
- ./plugins** non-SQL add-ons: display, style, row, arg default, arg validator, access, cache, query (3.x), exposed form (3.x), pager (3.x)
- ./theme** theme and preprocessor files
- ./modules** data models + specialized handlers
 - ./node**
 - ./user**
 - ./taxonomy**
 - ...

EXAMPLES // views_block

Exposing Drupal block data model to Views

hook_views_api()

inform Views that we exist (in mymodule.module)

hook_views_data()

define our data model (in mymodule.views.inc)

hook_views_handlers()

declare our field handlers (in mymodule.views.inc)

hook_views_default_views()

define our default views (in mymodule.views_default.inc)

handlers classes

define our field handlers (in individual files)

EXAMPLES // views_flipped_table

Provide new style plugin to render view results

hook_views_api()

inform Views that we exist (in mymodule.module)

hook_views_plugins()

declare our style plugin (in mymodule.views.inc)

theme template

as declared above (in separate file)

theme preprocessor

corresponding to theme template (in mymodule.module)

EXAMPLES // views_autosubmit

Auto-submit exposed form elements

hook_views_api()

not needed because we're bypassing Views!

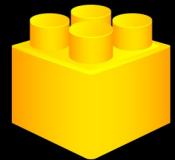
hook_form_views_exposed_form_alter()

form altering code for Views exposed filters (in mymodule.module)

Drupal.behaviors.views_autosubmit()

bind to form input change to auto-submit the form (in mymodule.js)

Thank you!



Karim Ratib (infojunkie@drupal)
There's a module for that!

<http://there is a module for that.com>