



## **Image Gallery Recipe**

**Workshop: Uschi Reiter**

# Übersicht

- **Kurze Beschreibung/Vorführung des Workshops-Ziels**

**<http://d6playground.servus.at>**

# Übersicht

- **Kurze Beschreibung/Vorführung des Workshops-Ziels**
- **Module**

# Module

Folgende Module werden hier benötigt um zu dem Ergebnis zu kommen: <http://d6playground.servus.at/>

## **Standardmäßige Module:**

Views

CCK

FileField, ImageField

ImageCache

## **Module extra gebraucht werden**

nodereference\_url

views\_attach

views\_gallery, feature (not working at the moment)

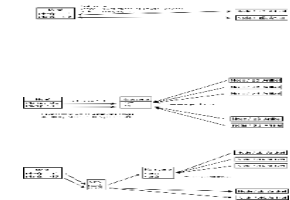
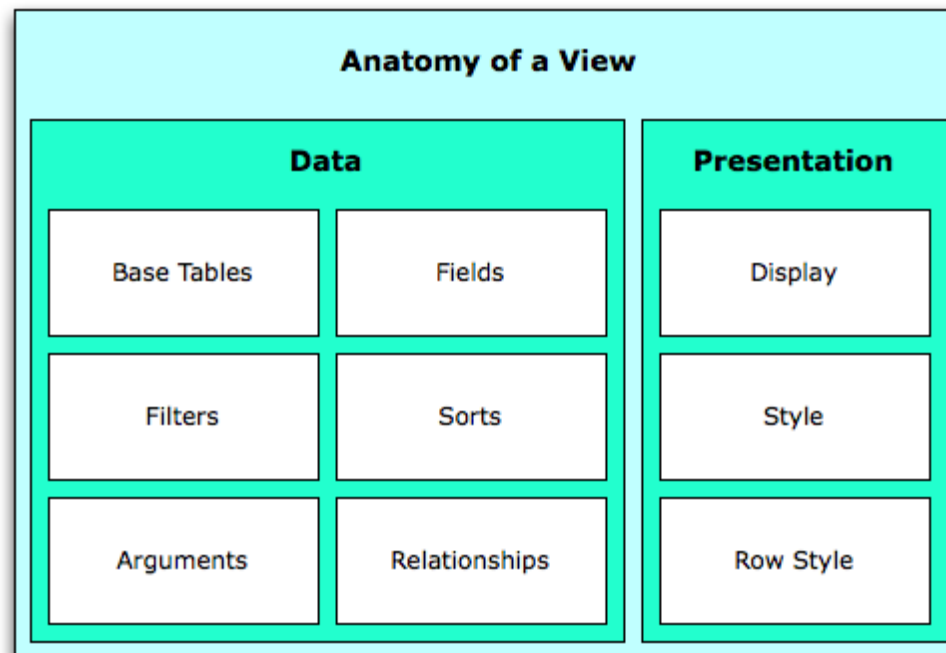
Optional: views\_carusel, jcarousel (für eine weitere Darstellungweise)

# Übersicht

- **Kurze Beschreibung/Vorführung des Workshops-Ziels**
- **Module**
- **Über Views (Wiederholung)**

# views

Mit Hilfe von „views“ kann man verschiedene Ansichten auf Inhalte in der Datenbank erstellen. Diese Ansichten und deren Inhalte können nach bestimmten Kriterien (zb: nach Taxonomy, Zeit, Autor, etc) gefiltert werden.



**Grafik Vergleich  
Taxonomy vs Views**

# views

**Die Ansicht so einer „view“ kann standardmäßig**

- Grid (Gitter, Raster)
- HTML List (Liste)
- Table
- Unformatted

Mit dem zusätzlichen Modul „views\_carusel, jcarousel“ können wir die Ansichten auf unsere Bilder erweitern.

# Übersicht

- **Kurze Beschreibung/Vorführung des Workshops-Ziels**
- **Module**
- **Über Views (Wiederholung)**
- **Praxisteil**
- **Image Gallery Recipe**



# Beginn praktischer Teil

## *Image Gallery Recipe*

# Praxis

## Runter laden der Module:

**noderefernce\_url:** [http://drupal.org/project/nodereference\\_url](http://drupal.org/project/nodereference_url)

**views\_attach:** [http://drupal.org/project/views\\_attach](http://drupal.org/project/views_attach)

**views\_carusel, jcarousel**

<http://drupal.org/project/viewscarousel>

<http://drupal.org/project/jcarousel>

Add: (noch nicht ganz schlau gweorden wie die beiden hier zum tragen kommen)

**views\_gallery:** [http://www.lullabot.com/files/views\\_gallery.zip](http://www.lullabot.com/files/views_gallery.zip) (?)

**Features:** <http://drupal.org/project/features>

**Module via FTP auf den Server laden**  
**/modules**

# Praxis

## Auf der Webseite folgende Module deaktivieren

die Module **Image**, **Image Assist**, **Image Gallery** deaktivieren wir, um nicht durcheinander zu kommen. Diese Module bieten im Prinzip ähnliche Funktionalitäten, wir aber wollen uns unsere eigene Gallery bauen und um Funktionalitäten/Features erweitern. Auch erstellen wir einen neuen Content Typen „Photo“ den wir mit einem File Field Image erweitern.

# Praxis

## **Folgende Module aktivieren**

nodereference\_url

views\_attach

views\_gallery

views\_carousel, jcarousel

## **weitere standardmäßig vorhandene Module**

ImageCache, ImageCache UI, ImageApi

# ImageCache

## Was ist ImageCache?

ImageCache is a dynamic image manipulation and cache tool. It allows you to create a namespace that corresponds to a set of image manipulation actions. It generates a derivative image the first time it is requested from a namespace until the namespace or the entire imagecache is flushed.

## Creating a Preset - Wir legen folgende Typen an

**photo\_thumbnail (200x200)-resize/crop**

**photo\_preview (300x200) - resize**

**photo\_big (400x 200) - rezise**

# Anlegen von 2 Content Typen

## **Add Content Type – Gallery**

Dieser Content Typ ist relativ einfach, das einzige was wir machen wir erlauben keine Kommentare, Attachments etc.

## **Add Content Type – Photo**

Wir brauchen hier kein Bodyfeld und deaktivieren ebenfalls, was wir nicht als notwendig erachten (Comments, etc)

# Einstellungen der Content Typen

**Type -Gallery** – hier werden so gut wie keine Einstellungen vorgenommen.

## Type – Photo

Home » Administer » Content management » Photos

Edit Manage fields Display fields

gallery | photo

### Photos

Add fields and groups to the content type, and arrange them on content display and input forms.  
Note: Installing the [Advanced help](#) module will let you access more and better help.

Label	Name	Type	Operations
Title	Node module form.		
Menu settings	Menu module form.		
photo	field_foto	File	<a href="#">Configure</a> <a href="#">Remove</a>
gallery	field_gallery	Node reference	<a href="#">Configure</a> <a href="#">Remove</a>

Add

**New field**

field\_

Label Field name (a-z, 0-9, \_) Type of data to store. Form element to edit the data.

Save

# Einstellungen der Content Typen

**Hinzufügen des FileField - Image mit dem Namen Photo**

**Einstellungen:** Eigentlich ist hier erstmal nur relevant

## **Global settings**

**X Required – Es muss ein Photo raufgeladen werden!**

**Number of values: 1**

Maximum number of values users can enter for this field.

'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.

## **List field:**

In unserem Fall **disabled**

## **Description field:**

in unserem Fall auch **off**



# Einstellungen der Content Typen

**Hinzufügen des Feldes: Noderreferenz** mit dem Namen **Gallery**

## **Einstellungen:**

Wir wollen in unserem Foto ja **auf eine Gallery verweisen** und zusätzlich auch noch einen **add Photo-Link** in jeder Gallery hinzufügen.

**Widget type:** Reference from Url

## **Photos settings**

**Fallback behavior: Use select list widget (=Nodereference Url)**

## **Referenceable node links:**

These settings will automatically make a link on nodes that can be referenced. Clicking the link will take the user to the new node form and prepopulate the value of this node reference field.

## **X Create link on the full view**

**Im Feld tragen wir „add Photo“ ein** (so nennen wir den Link)

**Return path:** referenced Node

**Global settings:** auf **Required** setzen!

**Content types that can be referenced:** Gallery!

# Erstellen der View

**Hier kommt das Modul `views_attach` zum tragen!**

# Erstellen diverser Views

**Administer->views -> add View** mit dem Namen: **gallery**

**Zuerst die Default Einstellungen für diese view**

Diese nimmt man vor um bestimmte Regeln, die bei der view „**gallery**“ gleich bleiben sollen **generell** fest zu legen.

# Default view

## Wichtige Bemerkungen:

**Basic Settings:** „Grid“ ist die Darstellung wie wir die Photos anzeigen lassen möchten.

**Filter:** Bestimmt was für Nodes wir aus unserem Pool an Inhalten in der Datenbank überhaupt anzeigen lassen wollen!

**Fields:** In Fields bestimme ich genau was für „Felder“ einer Node hier angezeigt werden sollen.

Wir wählen **Fields-Content-> photo\_thumbnail image linked to node.**  
(hier kommen unsere ImageCache Presets zum tragen)

**Argumente** kommen bei den Einstellungen wesentlich zum tragen!

# Default view

## Das Wichtigste:

**Arguments:** In unserem Fall müssen absolut notwendig hier auch noch **Content-Argumente** eingestellt werden.

Es sagt, dass wir im Url dieser view eine bestimmte **NodeID** bekommen , die zum **Typ Gallery** gehört. Und wenn wir das tun, dann wird diese view eben nur Nodes zeigen, die eine bestimmte Nodereferenzen haben, die auf diese Gallery zeigen.

# Default view settings

The screenshot shows the configuration page for a Drupal view named 'Defaults'. The page is organized into several sections:

- View settings:** Tag: None
- Basic settings:** Name: Defaults, Title: None, Style: Grid, Row style: Fields, Use AJAX: No, Use pager: Mini, Items per page: 6, More link: No, Distinct: No, Access: Unrestricted, Caching: None, Exposed form in block: No, Header: None, Footer: None, Empty text: None, Theme: Information
- Relationships:** None defined
- Arguments:** Content: gallery
- Fields:** Node: Title Title, Content: photo photo\_thumbnail image linked to node
- Sort criteria:** Node: Post date desc
- Filters:** Node: Published Yes

On the left side, there are navigation options: 'Full Gallery', 'Latest Image', a 'Page' dropdown, and buttons for 'Add display' and 'Analyze'.

# Default view

Nun sind wir mit dem Erstellen des **Default Views** fertig und brauchen einen **Display**. Mit Hilfe von „Views-Attach“ wollen wir diese View in den **Gallery Content Typen** einfügen.

# Display: Full Gallery

Add Display – **Node Content**

Name: Full Gallery

Neu sind die Einstellungen Node Content Settings

**Node content settings**

**Node types:** gallery

**Build modes:** Full node

**Arguments:** nid

**Show title:** No





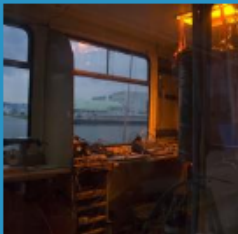



# Full Gallery View

**Meine erste Gallery**

[Home](#)

Submitted by [adm h](#) on Fri, 07/31/2009 - 14:37

<p>Title: xoxoxo</p>  <p>Post date: 4 days 23 min</p>	<p>Title: pickerl</p>  <p>Post date: 4 days 23 hours</p>
<p>Title: hunzi</p>  <p>Post date: 4 days 23 hours</p>	<p>Title: test</p>  <p>Post date: 4 days 23 hours</p>
<p>Title: Mein zweites Photo</p>  <p>Post date: 5 days 2 hour</p>	<p>Title: mein erstes photo</p>  <p>Post date: 5 days 2 hour</p>

[add Photo](#)

# Display: Full Gallery

The screenshot shows the configuration page for the 'Full Gallery' display. On the left, there are navigation tabs: 'Defaults', 'Full Gallery' (selected), and 'Latest Image'. Below these are controls for 'Page' (a dropdown menu), an 'Add display' button, and an 'Analyze' button. The main content area is titled 'Full Gallery' with the subtitle 'Display the view as part of a node's content'. It is divided into several sections: 'Basic settings', 'Relationships', 'Arguments', 'Fields', 'Sort criteria', and 'Filters'. The 'Basic settings' section includes fields for Name, Title, Style, Row style, Use AJAX, Use pager, Items per page, Distinct, Access, Caching, Exposed form in block, Header, Footer, Empty text, and Theme. The 'Node content settings' section at the bottom is highlighted with a red box and contains fields for Node types, Build modes, Arguments, and Show title. The 'Relationships' section shows 'None defined'. The 'Arguments' section shows 'Content: gallery'. The 'Fields' section lists 'Node: Title Title', 'Content: photo photo\_thumbnail image linked to node', and 'Node: Post date Post date'. The 'Sort criteria' section shows 'Node: Post date desc'. The 'Filters' section shows 'Node: Published Yes'. A 'Remove display' button is located at the top right.

Defaults

Full Gallery

Latest Image

Page

Add display

Analyze

Full Gallery *Display the view as part of a node's content*

**Basic settings**

Name: Full Gallery

Title: None

Style: Grid

Row style: Fields

Use AJAX: No

Use pager: Mini

Items per page: 6

Distinct: No

Access: Unrestricted

Caching: None

Exposed form in block: No

Header: None

Footer: None

Empty text: None

Theme: Information

**Node content settings**

Node types: gallery

Build modes: Full node

Arguments: nid

Show title: No

**Relationships**

*None defined*

**Arguments**

Content: gallery

**Fields**

Node: Title Title

Content: photo photo\_thumbnail  
image linked to node

Node: Post date Post date

**Sort criteria**

Node: Post date desc

**Filters**

Node: Published Yes

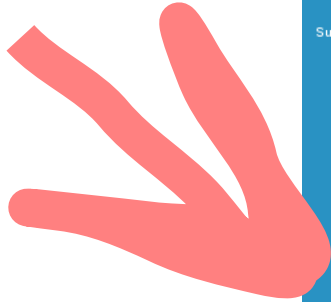
Remove display

# Display: Latest Image

Add Display – **Node Content**

Name: **Latest Image**

Damit wir in der Übersicht aller Gallerys immer das aktuellste Bild bekommen im Teaser angezeigt bekommen.



**meinte dritte Gallery**  
Submitted by [admin](#) on Fri, 07/31/2009 - 15:21




photo:

Vivamus eu leo vel orci tincidunt euismod. Maecenas posuere nulla eget urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus risus. Praesent eu metus. Suspendisse metus. Vestibulum quis nisl. Proin a sapien sit amet neque amet.

Vivamus eu leo vel orci tincidunt euismod. Maecenas posuere nulla eget urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus risus. Praesent eu metus. Suspendisse metus. Vestibulum quis nisl. Proin a sapien sit amet neque amet.

---

**meine zweiter gallery**  
Submitted by [admin](#) on Fri, 07/31/2009 - 14:44




photo:

Vivamus eu leo vel orci tincidunt euismod. Maecenas posuere nulla eget urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus risus. Praesent eu metus. Suspendisse metus. Vestibulum quis nisl. Proin a sapien sit amet neque amet.

# Display: Latest Image

The screenshot shows the configuration page for a Drupal display named 'Latest Image'. The interface is divided into several sections:

- Left sidebar:** Contains navigation links for 'Defaults', 'Full Gallery', and 'Latest Image'. Below these are a 'Page' dropdown menu, an 'Add display' button, and an 'Analyze' button.
- Main content area:** Titled 'Latest Image' with the subtitle 'Display the view as part of a node's content'. It contains two main sections:
  - Basic settings:** A list of configuration options including Name (Latest Image), Title (None), Style (Unformatted), Row style (Fields), Use AJAX (Yes), Use pager (No), Items to display (1), Distinct (No), Access (Unrestricted), Caching (None), Exposed form in block (No), Header (None), Footer (None), Empty text (None), and Theme (Information).
  - Node content settings:** A list of configuration options including Node types (gallery), Build modes (Teaser), Arguments (nid), and Show title (No).
- Right sidebar:** Contains three sections:
  - Relationships:** Shows 'None defined'.
  - Arguments:** Shows 'Content: gallery'.
  - Fields:** Shows 'Content: photo photo\_thumbnail image'.
- Top right:** A 'Remove display' button.
- Bottom right:** A 'Sort criteria' section showing 'Node: Post date desc' and a 'Filters' section showing 'Node: Published Yes'.

# Das Resultat



Da der Content Typ Gallery so eingestellt ist das er automatisch auf der Frontseite veröffentlicht wird erscheinen unsere Galerien jetzt auf der ersten Seite.

Natürlich kann auch eine eigene View erstellt und mit einem Pfad versehen werden wo nur die Galerie-Einträge gefiltert sind.

# Praxis Teil 2

**Wir Erstellen eine weitere „view“ mit views\_carusel**

# View Imagepage

**In dieser View lassen wir alle Bilder anzeigen-sprich  
Filtern unsere Inhalte nach Content Typen Photo**

# View Imagepage

The screenshot shows the configuration page for a Drupal view named 'imagepageblock'. The interface is organized into several sections:

- Defaults:** Shows the current view name 'imagepageblock' and a dropdown menu set to 'Page'. Below this are buttons for 'Add display' and 'Analyze'.
- View settings:** The 'Tag' is set to 'None'.
- Basic settings:** A list of configuration options including Name (Defaults), Title (None), Style (Grid), Row style (Fields), Use AJAX (No), Use pager (No), Items to display (Unlimited), More link (No), Distinct (No), Access (Unrestricted), Caching (None), Exposed form in block (No), Header (None), Footer (None), Empty text (None), and Theme (Information).
- Relationships:** Currently 'None defined'.
- Arguments:** Currently 'None defined'.
- Fields:** Lists 'Content: photo' with sub-fields 'photo\_thumbnail' and 'image linked to node'.
- Sort criteria:** Set to 'Node: Post date asc'.
- Filters:** Includes 'Node: Type = Photos' and 'Node: Published Yes'.



# Display Carousel

The image shows the configuration interface for a Drupal Views display named 'imagepageblock'. The display is currently set to 'Page' and is configured as a 'Views Carousel'. The configuration is divided into several sections:

- Defaults:** Shows the current display type as 'Page' and provides buttons for 'Add display' and 'Analyze'.
- Basic settings:** Includes fields for Name (imagepageblock), Title (None), Style (Views Carousel), Row style (Fields), Use AJAX (No), Use pager (No), Items to display (Unlimited), More link (No), Distinct (No), Access (Unrestricted), Caching (None), Exposed form in block (No), Header (None), Footer (None), Empty text (None), and Theme (Information).
- Block settings:** Includes Admin (None) and Caching (Do not cache).
- Relationships:** Currently 'None defined'.
- Arguments:** Currently 'None defined'.
- Fields:** Shows 'Content: photo' with sub-fields 'photo\_thumbnail' and 'image linked to node'.
- Sort criteria:** Set to 'Node: Post date' in ascending order.
- Filters:** Includes 'Node: Type = Photos' and 'Node: Published' set to 'Yes'.
- Actions:** A 'Remove display' button is visible at the top right.

# Jcarousel Css in style.css

```
/**
 * Carusel
 */
.jcarousel-skin-tango .jcarousel-container-horizontal {
    width: 600px;
    background: #2285B7 none repeat scroll 0 0;
    border: 1px dotted #fff;
}

.jcarousel-skin-tango .jcarousel-clip-horizontal {
    width: 600px;
    height: 200px;
}

.jcarousel-skin-tango .jcarousel-item {
    width: 200px;
    height: 200px;
}

/*Add Photo*/
ul.links.inline .photo_field_gallery {
    font-weight:bold;
    background:#2285B7;
    border: 1px dotted #000;}
```