

EW4

Failinė struktūra

 **apps** – sistemos aplikacijos

 **ew4** - aplikacija

 **site** - aplikacija

 **config** – bendra konfigūracija

 **data** – sistemos kuriami duomenys, smarty generuojami šablonai, įkeliami failai

 **protected** – per naršyklę nepasiekiami failai

 **public** – vieši failai, pasiekiami per naršyklę tiesiogiai

 **modules** – sistemos moduliai

 **static** – temos, js ir css bibliotekos

 **themes** – sistemoje naudojamos temos, tiek svetainėje, tiek administravime

 **vendor** – php paketai / bibliotekos / framework'ai. Kiekvienas gali turėti savo autoloaderį.

 **ew4**

 **smarty**

Sistemas veikimas: index.php

```
include 'vendor/ew4/autoload.php';

$core = new ew4\core\ew(array(
    'config'=>'config/ew4config.xml',
    'debug_level'=>include('config/define_debug.php')
));

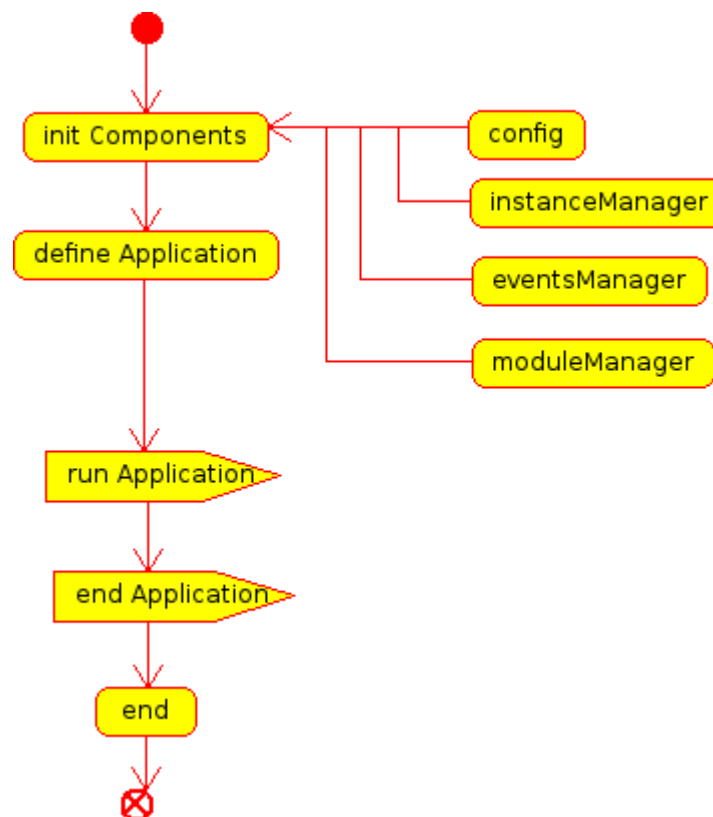
exit();
```

Sistemos veikimas: autoloader

- Kiekviena „vendor“ kataloge esanti biblioteka ar framework'as gali turėti savo individualų autoloaderį
- Bandant pakrauti biblioteką yra tikrinama ar bibliotekos kataloge egzistuoja autoload.php failas, Jei taip - jis įterpiamas
- EW4 autoload naudoja PHP spl_autoload funkcionalumą
 - Namespace = fizinis kelias
 - Papildomas ne root kelias yra vendor/ katalogas
 - Klasų failų plėtinys „.php“

Sistemas veikimas: ew objektas

- Inicializuoja
 - exceptionHandler
 - buffer
- Užregistruoja objektus
 - ew
 - config
 - request
 - app
 - events
 - modules
 - lang
- Paleidžia aplikacija
- Baigia procesa

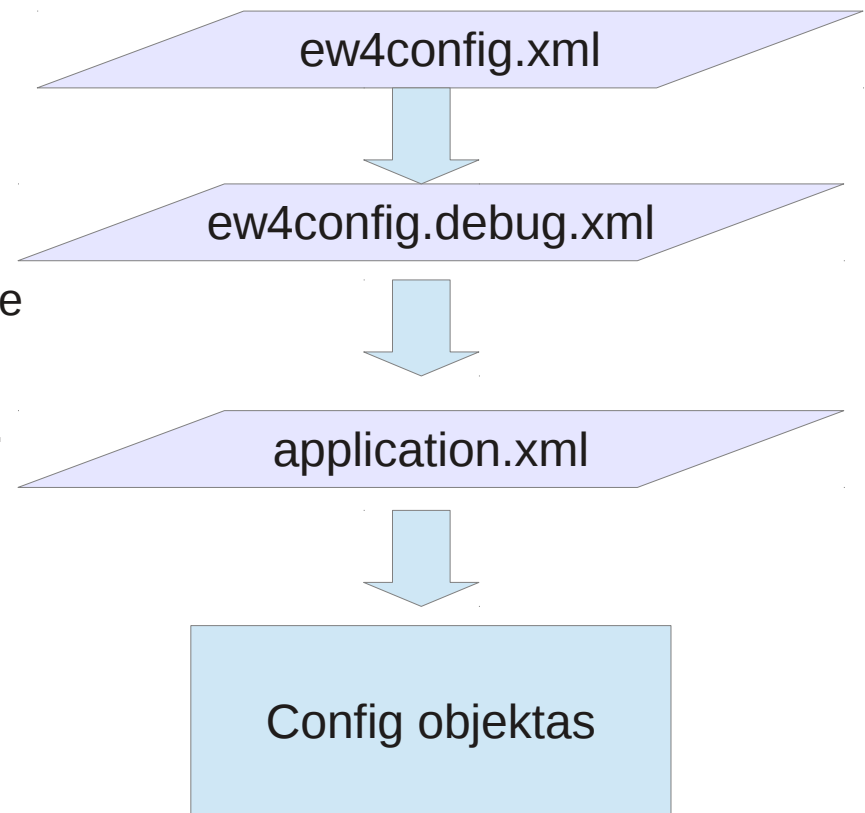


EW pagrindiniai “injectable” objektai

Objektas	Įterpimo vardas	Vieta	Pastabos
ew	ew	ew	
config	config	ew	
request	request	ew	
<i>Instance of application</i>	app	ew	Einamoji aplikacija
moduleManager	modules	ew	
lang	lang	ew	
eventsManager	events	ew	
db	db	app	
ew4Session	sess	app	
user	user	app	
page adminPage	page	app	Priklausomai nuo aplikacijos
<i>Instance of nodes/</i>	node	app	Priklausomai nuo aplikacijos. Site aplikacijoje visada.
notice	notice	app	
uilang	uilang	app	Tik ew4 aplikacijoje

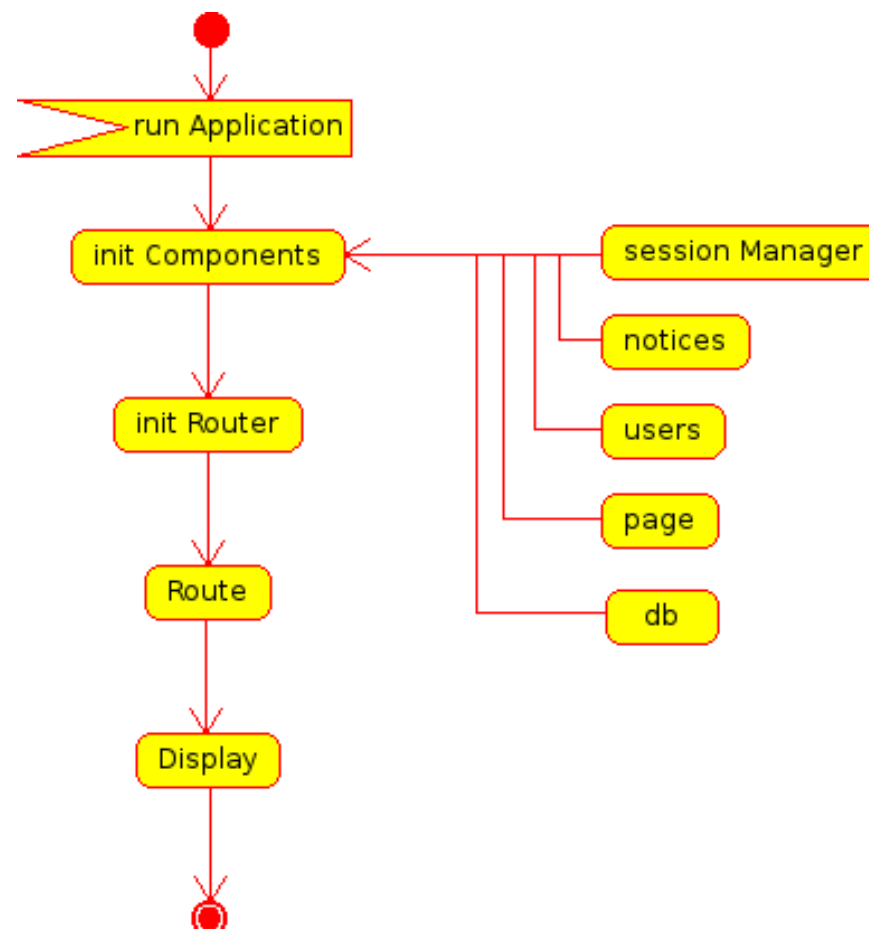
config objektas

- Nuskaito konfigūraciją iš failo
- Palaikomi failai
 - Php (*.ini.php, *.php)
 - Xml (*.xml)
- Konfigūracija saugoma ArrayObject tipo struktūroje kiekvienam gyliui
- Konfigūracijos pasiekiamumas bet kuriame gylyje. `$this->config->get(path)`, kur path
 - '*' arba '__root' - nuskaito viską
 - 'vienas.du.trys' – nuskaito kaip [vienas][du][trys]
- XML konfigūracijoje palaikomos funkcijos
 - import
 - act
 - return
 - remove
- update metodas



Application veikimas

- Užregistruoja objektus
- Inicializuoja routerį
- Vykdo rutinimą
- Atvaizduoja rutinimo rezultatą (view)



request objektas

- GET parametrai (paramsl)
- POST (paramsl)
- SERVER (paramsl)
- Puslapiavimo puslapis
- PATH (uriPathl)
- Vieši parametrai
 - lang (string)
 - app (string)
 - root_path (string)
 - root_url (string)
 - host (string)
- getQuery([parametras, [default]])
- getPost([parametras, [default]])
- getServer([parametas, [default]])
- getPage()
- getPath()
- Naudingi metodai
 - isPost()
 - isGet()
 - getPostJson()
 - issetQuery(parametras)
- Norint gauti visą POST masyvą
 - getPost()->getArray()

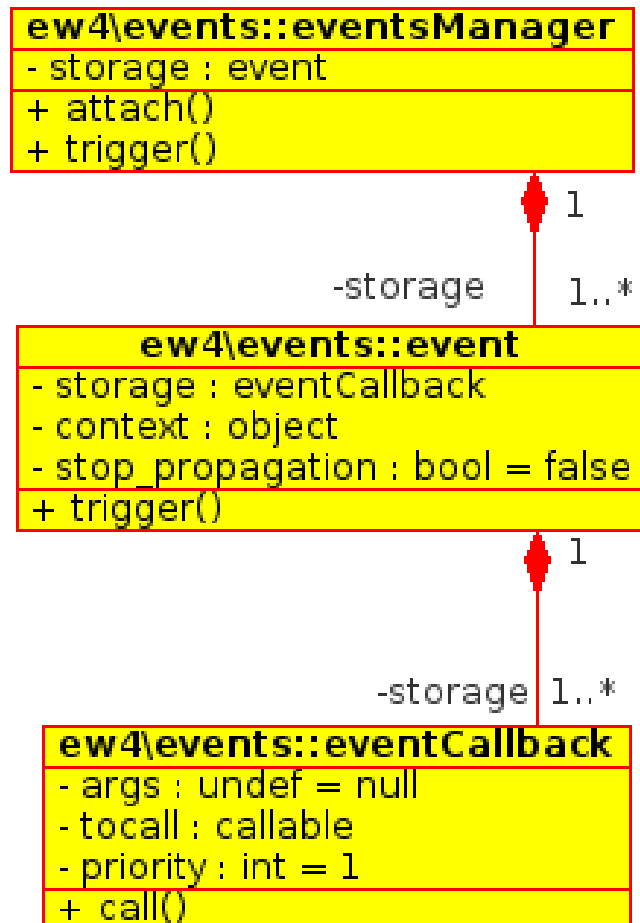
moduleManager

- Modulių nuskaitymas pagal config.php failus
- Kiekvienas modulis turi savo konteinerį “singleModule”
- Prašant modulio, moduleManager gražina modulio konteinerį
- singleModule konteineris
 - Validuoja modulio aprašymą
 - Iškviečia klases

```
->get('news')->loadModuleClass([klase, [konstruktoriaus argumentai<5]])
```

```
->get('news')->loadModuleClass() = newsModule
```

eventsManager



- Jei užregistruojama klasė, o ne objektas, metodas turi būti statinis
- Kiekvienas užregistruotas įvykio metodas gali sustabdyti tolimesnį vykdymą, `stopPropagation()`

lang ir uilang

- Kalbos parametrų objektai
- Suformuojami į konstruktorių paduodant dataArray tipo objektą gautą iš konfigūracijos
 - lang objektui „languages“
 - uilang objektui „ui_languages“
- Metodai
 - getLang()
 - getList()

ew4Session

- dataArrayWrapper tipo objektas
- Galioja visi parent klases metodai
- Kiti metodai
 - getId()
 - getName()
 - getUser()
 - setUser()
 - setExtra()
 - getExtra()

user

- Metodai
 - login()
 - logout()
 - (get|set)Parametras()

page objektas

- params tipo objektas
- Kaupiami atvaizdavimui skirti parametrai
- Savi metodai
 - set()
 - is_set()
 - getData()
 - updateOnEmpty()
- Kas naudoja?
 - Kontroleriai įrašo
 - Atvaizdavimas nuskaityto
- Administravime naudojamo platesnio objekto vidiniai objektai
 - menuObj (menuItem klasė)
- Administravimo objekto papildomi metodai
 - setActiveMenuObj()

notice objektas

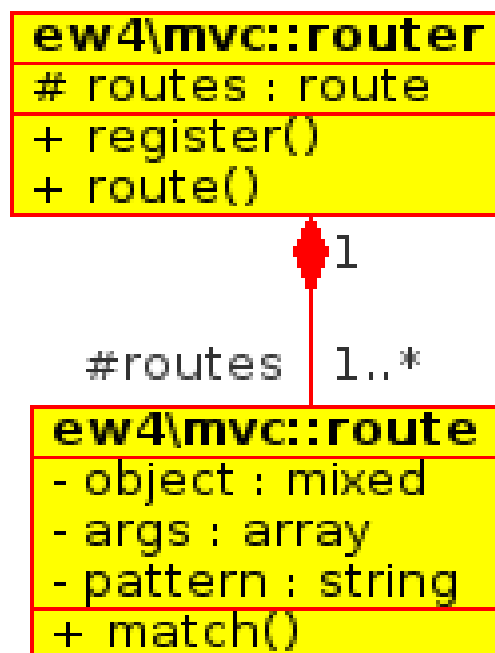
- Skirtas kaupti vartotojo atvaizdavimui skirtus pranešimus
 - Informacinio tipo
 - Klaidos tipo
- Metodai
 - setInfo()
 - getInfos()
 - setError()
 - getErrors()

node objektas

- Objektas priklauso nuo aplikacijos
 - site aplikacijoje naudojama runtimeNode klasė
 - ew4 aplikacijoje nodeItem klasės objektą įkrauna struktūros modulis
- Saugoma struktūros elemento informacija. Site aplikacijoje papildomai yra ir elemento route informacija.
- Galimi „magic“ metodai (get | set | isset)Parametras()
- Duomenų masyvas gaunamas metodu getData()

Sistemos veikimas: routinimas

- Naudojama „router“ klasė
- Routinimo taisyklės aprašomos reguliariosiomis išraiškomis
- Routinimo taisyklės gali būti taikomos tiek keliui (path) tiek GET argumentams
- Sutapimai iš reguliariosios išraiškos gali būti perduodami metodo argumentams
- Routeris nekviečia atitikusios taisyklės kontrolerio metodo
- route() grąžina „route“ objektą
- Gali perstumti uriPath objekto narius
- Gali turėti numatytąją routinimo taisyklę

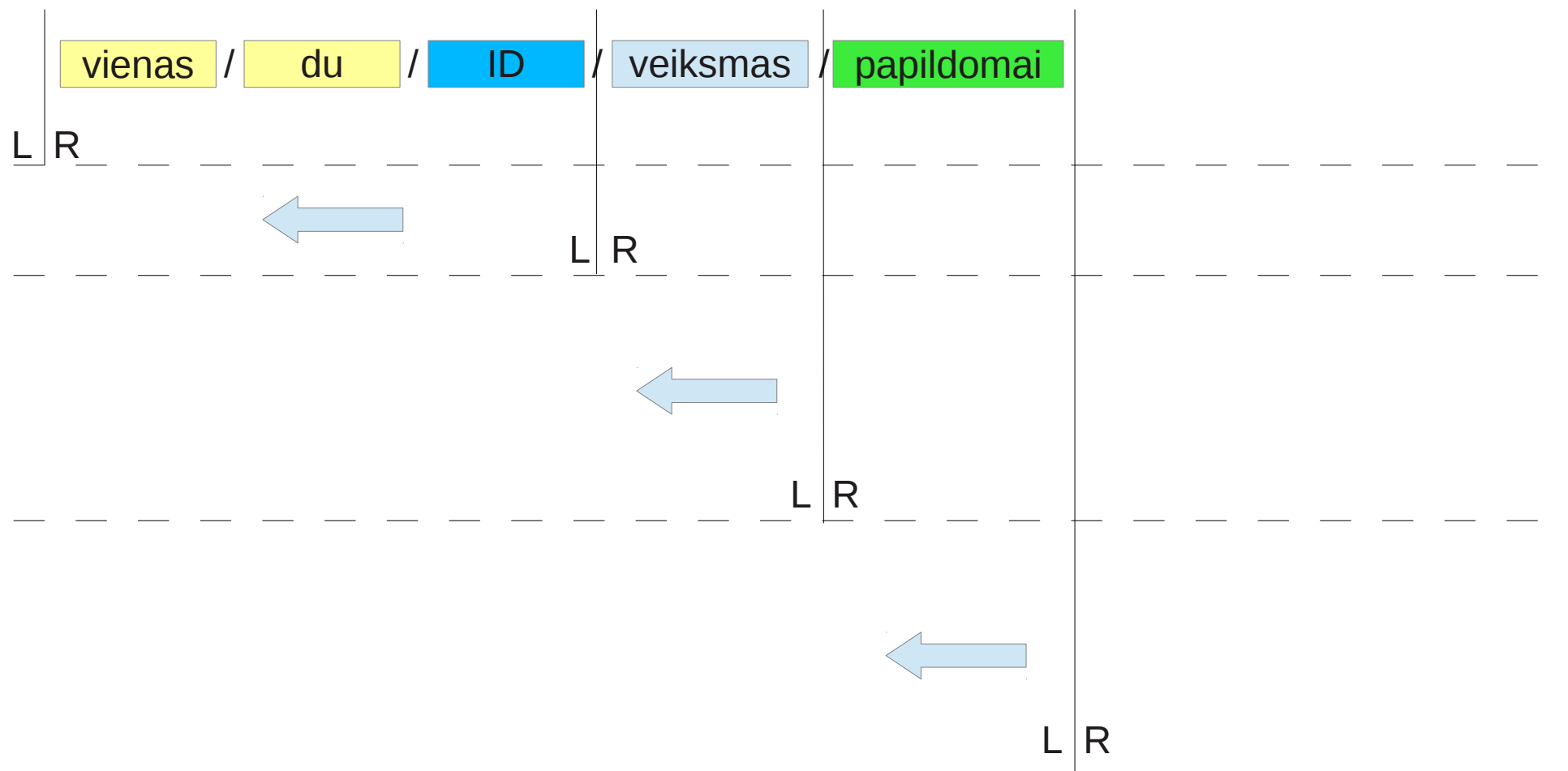


Sistemas veikimas: rutinimas, pavyzdžiai

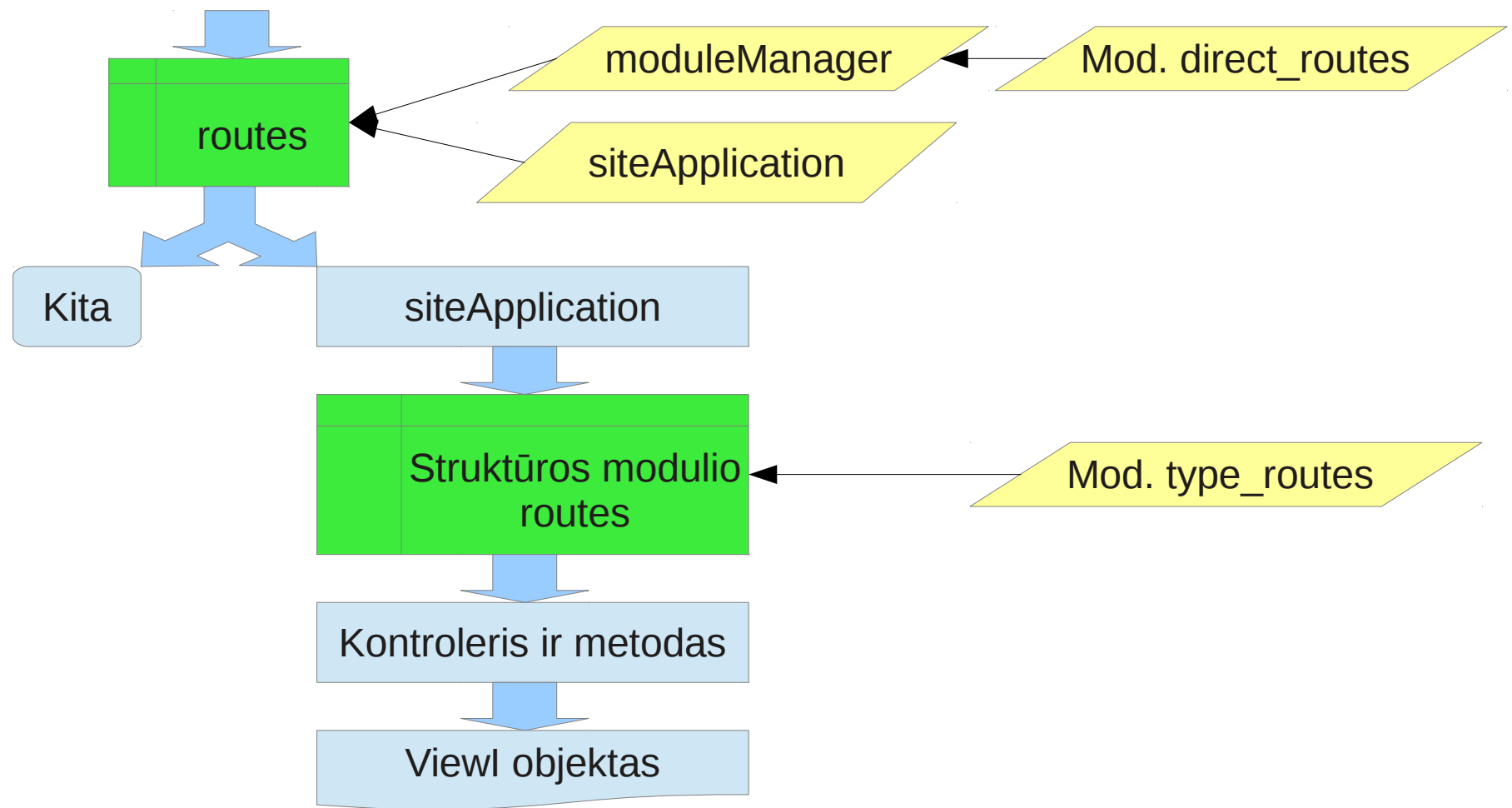
```
Array(  
  'controller'=>'apps\site\siteApplication',  
  'pattern' => '/^(?P<seourl>(([^\|/]+)\|)*)(>(?P<item>[0-9]+))(\|/$)/i',  
  'action' => 'detectControllerById',  
  'args' => array(  
    'seourl' => '',  
    'item' => 0  
  )  
) ,
```

```
Array(  
  'controller'=>'apps\ew4\ew4Application',  
  'pattern' => '/*./i',  
  'queryParams'=>array(  
    'switchUILang'=>null,  
  ),  
  'action' => 'changeUILang',  
  'dontshift' => true,  
  'args'=>array(  
    'switchUILang'=>''  
  )  
) ,
```

uriPath objektas

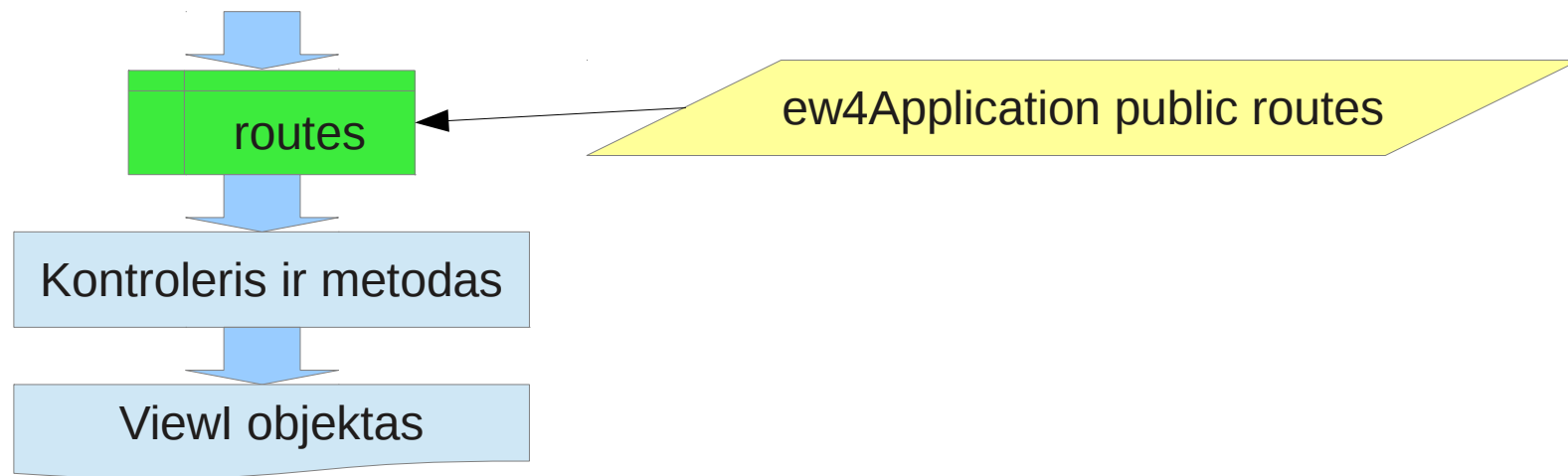


Sistemas veikimas: routinimas site aplikacijoje

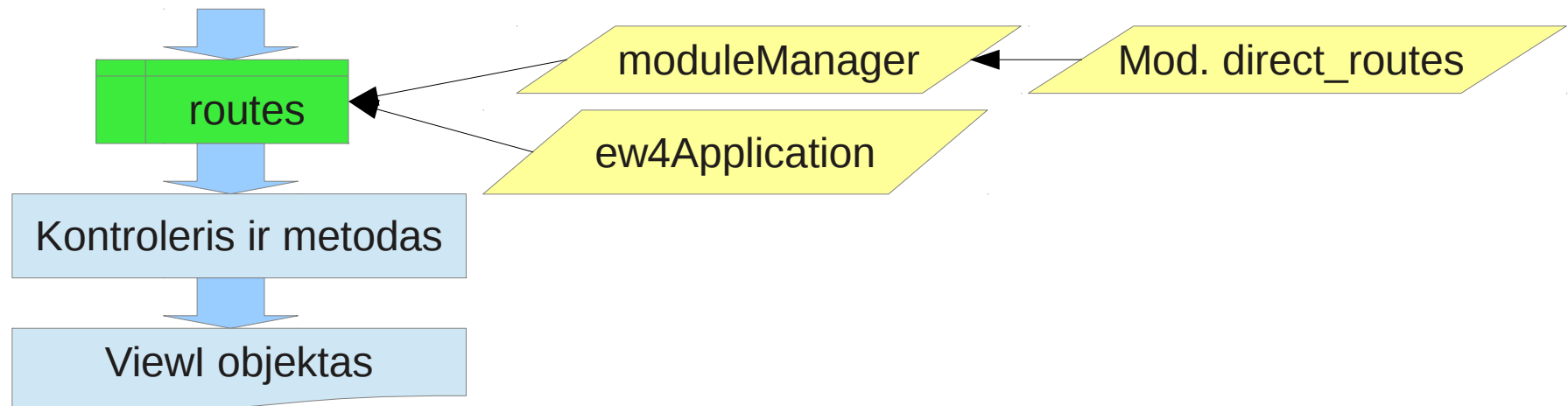


Sistemos veikimas: routinimas ew4 aplikacijoje

Neprisijungęs vartotojas



Prisijungęs vartotojas



Sistemos veikimas: display

- Išvedimui naudojami **view** tipo objektai
 - **pageView**
 - Skirtas atvaizduoti puslapio HTML
 - Praplečia „templateViewBase“
 - **jsonView**
 - Skirtas perduoti JSON tipo duomenis
 - Palaiko JSONP
 - **modalView**
 - Skirtas atvaizduoti HTML modaliniuose languose (ew4)
 - Praplečia „templateViewBase“
 - Naudoja kitas konfigūracijas nei pageView
 - **dialogView**
 - Skirtas atvaizduoti kitokios konstrukcijos HTML
 - praplečia pageView pakeisdamas pagrindinį šabloną į 'dialog.tpl'
- Atvaizdavimas vyksta kviečiant display() metodą
- Išveda Debug informaciją

templateViewBase klasės paveldėtojai

- Paskirtis išvesti HTML
- Naudoja Smarty
- Trigerina įvykius: pageCreated, onPageRender, afterPageRender, beforePageOutput, afterPageOutput
- Gali turėti papildomą service klasę įvykiams pagal temą apdoroti
- Palaiko kolektorius
- Automatiškai registruoja CSS, JS kolektorių bibliotekas aprašytas konfigūracijoje

pageView

- Į konstruktorių galima paduoti šių parametrų masyvą:
 - „libs“ norimos užregistruoti į kolektorius bibliotekos – bibliotekų vardų masyvas
 - „vars“ šablono kintamųjų vertės. Pagrindiniai šablono kintamieji
 - „content_template“ - turinio šablonas
 - „content“ - turinys arba jo verčių masyvas
 - „collectors“ reikalingi kolektoriai – asociatyvus masyvas
 - „basetemplate“ pagrindinis išvedimo šablonas – jei netinka 'index.tpl'
- Metodai
 - setBaseTpl() priskiria pagrindinį šabloną
 - assignVar() priskiria kintamąjį tiesiogiai
 - loadLib() užregistruoja biblioteką
 - collect() priskiria kolektoriui

pageViewService

- Iškviečiamas pageView konstruktoriuje
- Atlieka aplikacijai ar jos temai reikalingus veiksmus
 - Suformuoja meniu
 - Priskiria puslapiavimo kintamųjų vertes
 - Priskiria bendrųjų sistemos kintamųjų vertes
 - Priskiria kalbos vertes
 - Priskiria notice pranešimus
- Naudoja siteInfo servisą
- Po Smarty šablonų sugeneravimo, `DEBUG_LEVEL>1` , gali išvalyti užkešuosius Smarty duomenis: GET parametras „clear_tpl=on“
- Šablono kintamieji
 - ROOT_URL
 - ROOT_PATH
 - THEME_PATH
 - THEME_URL
 - SITEINFO : pavadinimai, bendri tekstai (pvz. „puslapis nerastas“)
 - PAGE : meniu, kalbos, puslapiavimas
 - errors
 - infos

Vaizdavimo konfigūracija: smarty

```
<smarty>
```

```
    <dir>vendor/smarty/</dir>
```

```
    <use_sub_dirs type="bool">1</use_sub_dirs>
```

```
    <direct_access_security type="bool">0</direct_access_security>
```

```
    <debugging type="bool">0</debugging>
```

```
    <compile_check type="bool">1</compile_check>
```

```
    <caching type="bool">1</caching>
```

```
    <force_cache type="bool">1</force_cache>
```

```
    <cache_lifetime type="int">86400</cache_lifetime>
```

```
    <compile_dir return="ew4\std\fn::combine" first_arg="DATA_DIR" >
```

```
protected/smarty_site_compile</compile_dir>
```

```
    <cache_dir return="ew4\std\fn::combine" first_arg="DATA_DIR" >
```

```
protected/smarty_site_cache</cache_dir>
```

```
</smarty>
```

Vaizdavimo konfigūracija: tpl

```
<tpl>
  <collectors>
    <jsvars>string</jsvars>
    <jslibs>file</jslibs>
    <js>file</js> ...
  </collectors>
  <libs>
    <fancybox2>
      <jslibs>
        <file1>static/libs/fancybox2/source/jquery.fancybox.js</file1>
      </jslibs>
      <csslibs>
        <file1>static/libs/fancybox2/source/jquery.fancybox.css</file1>
      </csslibs>
      <require>
        <lib1>jquery</lib1>
      </require>
    </fancybox2> ...
  </libs>
  <autolib>
    <fancybox2>1</fancybox2> ...
  </autolib>
  <autoload>
    <css>
      <file1>static/themes/site/css/base.css</file1>
      <file2>static/themes/site/css/style.css</file2>
    </css>
  </autoload>
  <service>
    <pageview>
      <class>\themes\site\php\pageViewService</class>
    </pageview>
  </service>
</tpl>
```

Vaizdavimo konfigūracija: themes

```
<themes>
  <default>site</default>
  <list>
    <site>
      <dir>static/themes/site/</dir>
      <tpl>
        <autolib> ... </autolib>
      </tpl>
      <smarty> ... </smarty>
    </site>
    <mobile>
      ...
    </mobile>
  </list>
</themes>
```

Dependency injection

- Priskiriama per public parametrus. Jei nurodytu public parametro vardu instanceManager turi objektą ir parametras nėra nustatytas – jam priskiriamas objektas
- „__require“ - public parametras kuriame išvardinami būtini objektai. Jei instanceManager tokio objekto neturi generuojamas instanceException
- „__inject“ - public parametras kuriame išvardinami noriki prijungti objektai. Tokiu atveju priskiriami tik išvardinti ir o ne visi public parametrai. Kiekvienas išvardintas objektas turi tureti atitinkamą public parametną klasėje
- Galimas priskyrimas į privačius parametrus „privateInject()“ metodo pagalba „target“ klasėje (injectableTrait šį metodą pateikia)
- Galimas priverstinis parametrų sukūrimas pagal priskiriamus objektus

Moduliai

- Modulių vykdytys failai talpinami kataloge, kurio pavadinimas atitinka modulio vardą. Katalogas talpinamas į „modules/“
- Šablonų failai talpinami modulio vardu pavadintame kataloge, temos šablonų kataloge (pvz.:
static/themes/site/tpl/news/news_index.tpl)
- Javascript failai talpinami analogiškai - static/themes/{theme}/js/
- CSS failai - static/themes/{theme}/css/

Moduliai: failinė struktūra

- **modules/mv/**
 - **models/** - modulio modeliai
 - **mv**elementmodel.php – elemento modelis
 - **mv**listmodel.php – elementų sąrašo modelis
 - **config.php** – modulio aprašymas / konfigūracija
 - **mv**module.php – pagrindinė modulio klasė
 - **mv**sitedefaultcontroller.php – site aplikacijos kontrolieris
 - **mv**ew4defaultcontroller.php – ew4 aplikacijos kontrolieris
 - **mv**events.php – įvykių apdorojimo klasė
 - **mv**callbacks.php – callback klasė

Moduliai: config.php

```
return array(  
    'id'           => 'news',  
    'active'       => true,  
    'version'      => '1.0.5',  
    'title'        => array(  
        'lt' => 'Naujienos'  
        'en' => 'News'  
    ),  
    'icons'        => array(  
        'main'=>'fa-file-text'  
    ),  
    'controllers' => array(...),  
    'settings'     =>array(  
        'itemsperpage'=>10,  
        ...  
    ),  
    'callbacks'    => array( ... ),  
    'events'       => array( ... )  
);
```

Moduliai: config.php controllers

```
array(  
    'site' => array(  
        'actions' => array(  
            'preview'=>array('access'),  
        ),  
        'type_routes' => array(  
            'news' => array([... RULES ...]),  
            'single' => array([... RULES ...]),  
            'default' => array([RULE]),  
        )  
    ),  
    'ew4' => array(  
        'direct_routes' => array([... RULES ...]),  
        'structure_routes' => array([... RULES ...])  
    )  
),
```

Moduliai: config.php controllers

```
array(  
    'pattern'=>'/^structure\/menu(?:P<menu>[^\\/]+)?\/*$\/i',  
    'action'=>'index_{menu}',  
    'args'=>array(  
        'menu'=>null  
    ),  
    'options' => array(  
        'page' => array(  
            'translate:title' => array(  
                'lt'=>'Svetainės struktūra',  
                'en'=>'Site structure'  
            )  
        ),  
        'tpl'=>array(  
            'vars'=>array(  
                'content_template' => 'structure/kitoks.tpl'  
            )  
        )  
    ),  
    'activemenu' => 'structure.{menu}'  
),
```

Moduliai: config.php

events

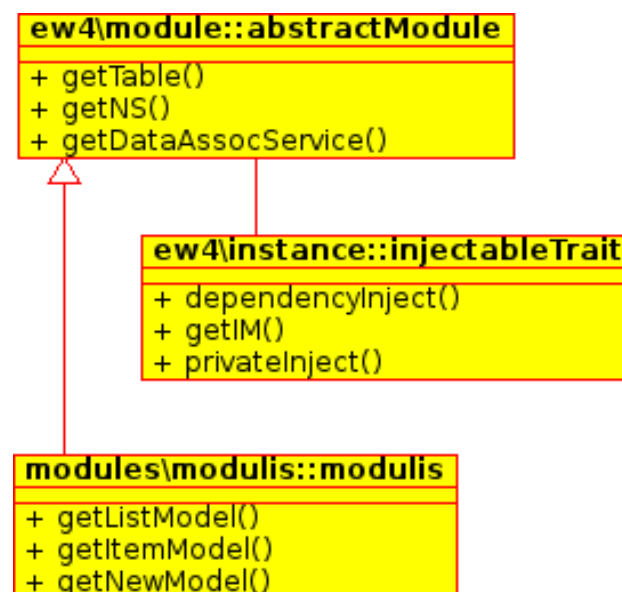
```
array(  
    'bootstrap' => array(  
        'onBootstrap' => array(  
            'call' => array(  
                'modules\titlepage\titlepageEvents',  
                "onBootstrap"  
            )  
        )  
    ),  
    'site' => array(  
        'loadDefaultPage' => array(  
            'call' => array(  
                'modules\titlepage\titlepageEvents',  
                "loadDefaultPage"  
            ),  
            'priority'=>1  
        ),  
    ),  
    'ew4' => array(  
        'initMenu' => array(  
            'call' => array(  
                'modules\titlepage\titlepageEvents',  
                "registerEw4Menu"  
            )  
        ),  
    )  
)
```

Moduliai: config.php callbacks

```
array(  
    'site' => array(  
        'assocContentGroup' => array(  
            'call' => array('newsCallbacks', 'assocItems'),  
            'args' => array('group_by_node' => true),  
        )  
    ),  
    'ew4'=>array(  
        'assocContentGroup' => array(  
            'call' => array('newsCallbacks', 'assocEditItems'),  
            'config'=>array(  
                'template' => 'img_date_title'  
            )  
        ),  
        'assocSearch' => array(  
            'call' => array('newsCallbacks', 'assocSearch')  
        )  
    )  
),
```

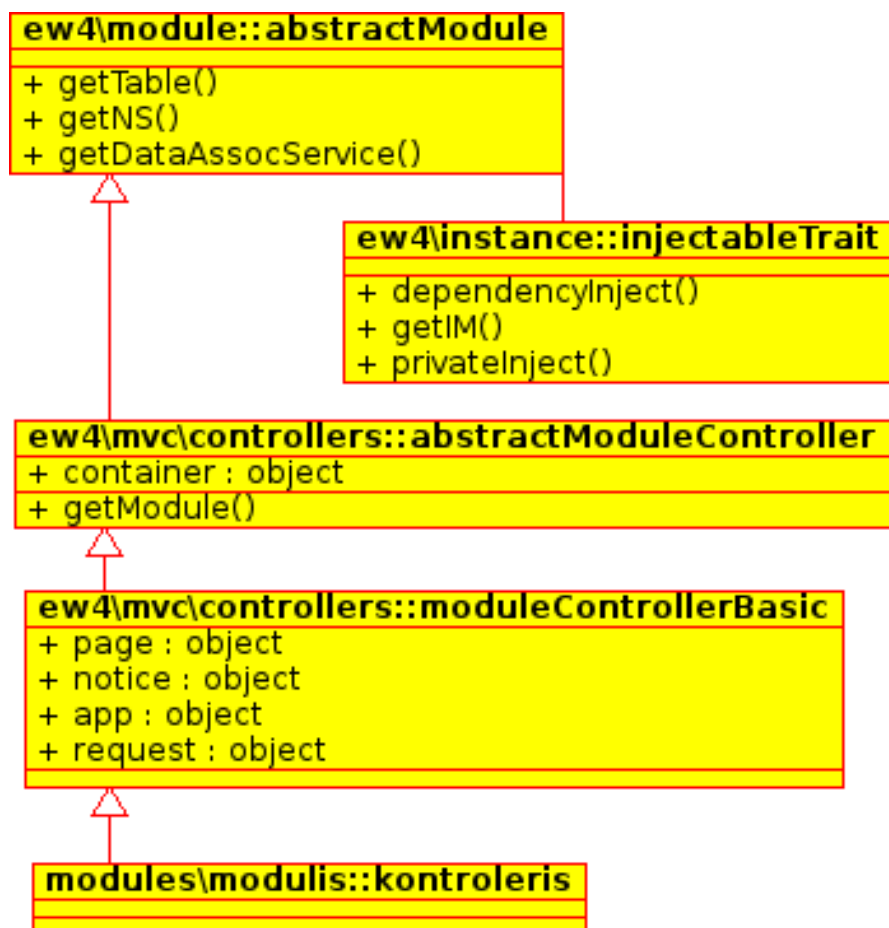
Moduliai: modulio klasė

```
namespace modules\quotes;  
  
use ew4\module\abstractModule;  
  
class quotesModule extends abstractModule  
{  
    protected $tb = array(  
        'items' => 'mod_quotes',  
    );  
  
    public $container;  
    ...  
}
```



- Dependency injection vykdomas konstruktoriuje, paveldėta iš **abstractModule**
- **\$container** – einamojo modulio konteinerinė klasė „**singleModule**“. Jei yra public parametras - prijungiama automatiškai užkraunant modulio klasę.

Moduliai: kontrolerio klasė



- Kontroleryje aprašomi rutinami metodai
- Kontrolerio rutinamas metodas visada turi grąžinti viewl objektą arba peradresuoti
- Rutinimo taisyklėje aprašyti argumentai bus perduoti kaip metodo argumentai
- moduleControllerBasic klasė automatiškai nustato šablono kintamojo „content_template“ vertę: {modulis}/{modulis}_{metodas}.tpl
- Reaguoja į rutinimo taisyklės „options“ parametą

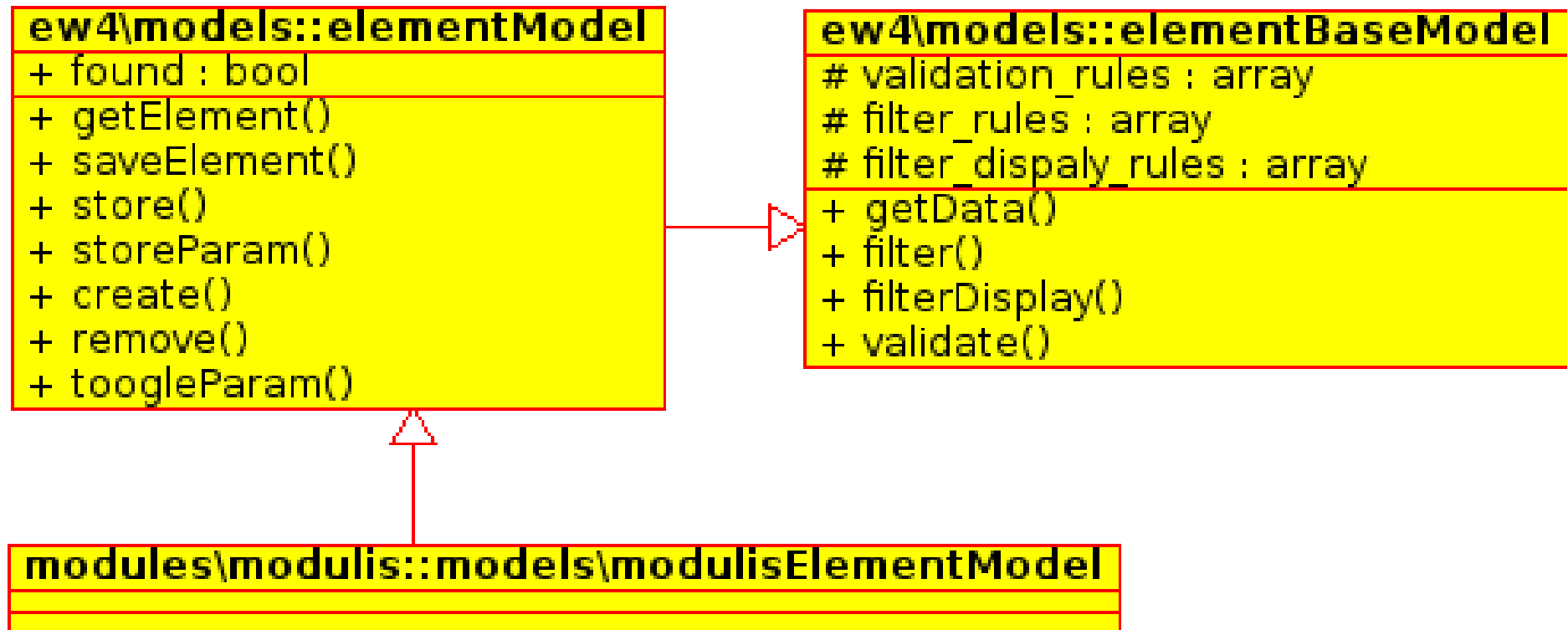
Moduliai: kontrolerio klasė

```
function index(){
    $url = new url(array(
        'seo_path'=>$this->node->getNodepath(),
        'path'=>array($this->node->getRouteId()
    ));
    $model = $this->getModule()->getListModel(array(
        'active'=>1,
        'node_id'=> (int) $this->node->getNodeId(),
    ));
    $lim = $this->container->get('settings.itemsperpage');
    $offset = $this->request->getPage();
    $r = $model->getList(array('limit'=>$lim, 'offset'=>$offset));
    $this->page->set('pager', new pager(array(
        'total'=>$model->totalRows,
        'limit'=>$lim,
        'offset'=>$offset,
        'url'=>$url->getString(null, array('p%d'))
    )));
    $this->setTPContent($r);
    return new pageView($this->getTP());
}
```

Modulio modeliai: *ListModel

```
namespace modules\news\models;
use ew4\models\ListModel;
class newsListModel extends ListModel
{
    protected $default_model_name = 'newsElementModel';
    function __construct(){
        parent::__construct();
        $this->default_element_model = __NAMESPACE__.'\\'.$this->default_model_name;
    }
    public function processListResults($results, $params = null){
        $this->registerNewsListProccesors($params);
        return parent::processListResults($results, $params);
    }
    public function registerNewsListProccesors($params){
        $this->attachEvent('listElementProcess',
            array(&$this, 'listElementProcess'), $params);
        $this->attachEvent('listElementProcess',
            array(&$this, 'processAssocStat'));
        $this->attachEvent('listElementProcess',
            array(&$this, 'callDisplayFilter'), null, 99);
        $this->attachEvent('postListElementProcess',
            array(&$this, 'listElementToArray'), null, 99);
        if(isset($this->module) && isset($this->sizeslist)){
            $this->attachEvent('preListElementProcess',
                array(&$this, 'assignSizesParams'));
        }
    }
}
...
```

Modulio modeliai: *ElementModel



Modulio modeliai: *ElementModel

- Veikia „magic“ metodai
 - setParametras()
 - getParametras()
 - issetParametras()
 - unsetParametras()
 - validateParametras()
 - filterParametras()
 - dfilterParametras()

Kiti modeliai

- globalstorage
- siteInfo
- nodeItem
- routeItem

DB

- Yra lentelių su išoriniais raktais
- Numatytas lentelių tipas: InnoDB
- Naudoja PDO
- Užklausos daromos naudojant prepare statement

Taisyklės

- Failų vardai mažosiomis
- Klasių ir metodų varduose kiekvienas loginis žodis pradedamas iš didžiosios, pirmas rašomas mažąja.
- Klasių varduose nenaudojame „_“
- Kiekviena klasė privalo turėti namespace
- Klasės namespace struktūra atitinka fizinį kelią
- Klasių kintamieji ir metodai turi turėti tipą: public, private, protected ir t.t.
- Metoduose nenaudoti daugiau 4 argumentų
- Šablonų failai pagal šabloną: {modulis}/{modulis}_{metodas}.tpl