

# Yii2: Displaying Data

Oleh: Ahmad Syauqi Ahsan



# Data Formatter

- Untuk memudahkan pengguna dalam membaca data, Yii menyediakan class `yii\i18n\Formatter`.
- Class ini dapat diakses melalui `Yii::$app->formatter`.
- Class formatter ini dapat digunakan untuk memformat tanggal, nilai angka, prosentase, dan lain-lain.

```
echo Yii::$app->formatter->asDate('2014-01-01', 'long'); // output: January 1, 2014
echo Yii::$app->formatter->asPercent(0.125, 2); // output: 12.50%
echo Yii::$app->formatter->asEmail('cebe@example.com'); // output: <a
href="mailto:cebe@example.com">cebe@example.com</a>
echo Yii::$app->formatter->asBoolean(true); // output: Yes

echo Yii::$app->formatter->format('2014-01-01', 'date'); // output: January 1, 2014
// you can also use an array to specify parameters for the format method:
// `2` is the value for the $decimals parameter of the asPercent()-method.
echo Yii::$app->formatter->format(0.125, ['percent', 2]); // output: 12.50%
```

# Date Formatter with Locale

- Formatter ini juga akan menyesuaikan dengan application property "language" atau property "locale" pada komponen Formatter.
- Misal, untuk format tanggal, nama hari dan bulan akan disesuaikan dengan bahasa yang dipilih.
- Syarat: PHP Extension "intl" harus aktif.
- Note: keluaran dapat berbeda untuk versi library "ICU" yang berbeda. Pastikan menggunakan versi library "ICU" yang sama untuk semua lingkungan server anda.

```
Yii::$app->formatter->locale = 'en-US';  
echo Yii::$app->formatter->asDate('2014-01-01'); // output: January 1, 2014  
Yii::$app->formatter->locale = 'de-DE';  
echo Yii::$app->formatter->asDate('2014-01-01'); // output: 1. Januar 2014  
Yii::$app->formatter->locale = 'ru-RU';  
echo Yii::$app->formatter->asDate('2014-01-01'); // output: 1 янiàðÿ 2014 ã.
```

# Konfigurasi Formatter

- Anda dapat mengubah nilai default dari Formatter ini melalui file konfigurasi dari aplikasi (misal: "web.php" untuk basic template). Perhatikan contoh konfigurasi disamping.
- Nilai default tersebut dapat di-overwrite melalui properti pada pemanggilan metode (perhatikan contoh dibawah).

```
// .. Other configurations
'components' => [
  // .. Other components
  'formatter' => [
    'dateFormat' => 'dd MMM yyyy',
    'decimalSeparator' => ',',
    'thousandSeparator' => ' ',
    'currencyCode' => 'EUR',
  ]
]
// ...
```

```
echo Yii::$app->formatter->asDate('2014-01-01'); // output: 01 Jan 2014
echo Yii::$app->formatter->asDate('2014-01-01', 'long'); // output: January 1, 2014
echo Yii::$app->formatter->asDate('2014-01-01', 'd MMMM yyyy'); // output: 1 January 2014
```

# Memformat Tanggal dan Waktu

- Class `Formatter` memiliki beberapa metode berbeda untuk memformat nilai tanggal dan waktu.
  - `date` – tampilan tanggal. Contoh: January 01, 2016
  - `time` – tampilan waktu. Contoh: 14:23
  - `datetime` – tampilan tanggal dan waktu. Contoh: January 01, 2014 14:23
  - `timestamp` – tampilan timestamp di Unix. Contoh: 1412609982
  - `relativeTime` – tampilan interval waktu antara nilai tanggal dengan waktu saat ini. Contoh: 1 hour ago.
- Format dari `date`, `time`, dan `datetime` dapat diatur secara global melalui konfigurasi komponen `Formatter` pada property `$dateFormat`, `$timeFormat`, dan `$datetimeFormat`.
- Nilai pada property diatas dapat berupa "short", "medium", "long", dan "full". Anda juga dapat mengatur lebih detail sesuai dengan aturan dari library ICU (<http://userguide.icu-project.org>).

# Memformat Angka

- Untuk memformat angka, class `Formatter` menyediakan metode-metode berikut:
  - `integer` – tampilan angka bulat (nilai decimal akan di-truncate). Contoh: 42.
  - `decimal` – tampilan angka decimal. Contoh: 2,542.123 atau 2.542,123.
  - `percent` – tampilan angka prosentase. Contoh: 42%.
  - `scientific` – tampilan angka scientific. Contoh 4.2E4.
  - `currency` – tampilan nilai mata uang. Contoh: \$420.00  
Catatan: agar fungsi ini bekerja dengan benar, nilai property "locale" harus mengandung nilai untuk negara. Contoh: "id\_ID", "en\_GB", "en\_US", dll.
  - `size` – tampilan angka bytes dengan format yang mudah untuk dibaca. Contoh: 410 kibibytes, 3 mebibytes, dll.
  - `shortSize` – seperti format "size" namun dalam bentuk pendek. Contoh: 410 KiB, 3 MiB, dll.

```
echo Yii::$app->formatter->asInteger('42.71'); // output: 42
echo Yii::$app->formatter->asDecimal('2542.123'); // output: 2,542.123
echo Yii::$app->formatter->asShortSize('419840'); // output: 410 KiB
```

# Format Lainnya

- Selain untuk memformat tanggal dan angka, class `Formatter` dapat digunakan untuk memformat nilai: `raw`, `text`, `ntext`, `paragraphs`, `html`, `email`, `image`, `url`, dan `Boolean`.
- Untuk nilai `null` (kosong), `Formatter` akan menampilkannya sebagai sebuah placeholder yang secara default bernilai `"(not set)"`. Kata `"not set"` ini akan ditampilkan sesuai dengan konfigurasi Bahasa.
- Anda dapat mengkonfigurasi property `"nullDisplay"` untuk mengatur custom placeholder.

# Pagination

- Ketika terdapat banyak data yang ingin ditampilkan dalam beberapa halaman → dibutuhkan `Pagination` (`yii\data\Pagination`).
- Property "totalCount" harus diisi dengan jumlah data keseluruhan.
- Untuk mengatur jumlah data per halaman dapat dilakukan melalui property "pageSize".
- Jika anda menggunakan Data Provider dan salah satu dari Data Widgets, `Pagination` secara otomatis akan aktif.

```
// in the Controller
function actionIndex()
{
    $query = Article::find();
    $pages = new \yii\data\Pagination([
        'totalCount' => $query->count(),
        'pageSize' => 10
    ]);
    $models = $query->offset($pages->offset)
->limit($pages->limit)->all();

    return $this->render('index', [
        'models' => $models, 'pages' => $pages,
    ]);
}
```

```
// in the View
// ...
foreach ($models as $model) {
    // display the $model data here
}

// display pagination
echo \yii\widgets\LinkPager::widget([
    'pagination' => $pages,
]);
// ...
```



# Pengurutan (Sorting)

- Yii menyediakan class `yii\data\Sort` untuk menangani Sorting.
  - Buat objek dengan class `Sort`
  - Tambahkan atribut yang akan digunakan untuk pengurutan (dalam array) pada property "attributes".
  - Gunakan objek tersebut pada pembuatan models data (masukkan pada "orderBy").
  - Lewatkan objek tersebut ke View bersamaan dengan models.
  - Tampilkan link untuk pengurutan pada View.
- Jika anda menggunakan Data Provider dan salah satu dari Data Widgets, Sorting secara otomatis akan aktif.

```
// in the Controller
function actionIndex()
{
    $sort = new \yii\data\Sort([
        'attributes' => [
            'age',
            'name'
        ]
    ]);
    $models = Article::find()
        ->where(['status' => 1])
        ->orderBy($sort->orders)
        ->all();
    return $this->render('index', [
        'models' => $models,
        'sort' => $sort,
    ]);
}
```

```
// in the View
// ...
echo $sort->link('name') . " | "
    $sort->link('name');

foreach ($models as $model) {
    // display the $model data here
}
// ...
```

# Data Providers

- Data Providers merupakan abstraksi untuk data lengkap dengan pengelolaan Pagination dan Sorting.
- Data Providers dapat digunakan untuk grids, lists, dan data widget yang lain.
- Yii menyediakan 3 macam data providers yaitu:
  - `yii\data\ActiveDataProvider`,
  - `yii\data\ArrayDataProvider`, dan
  - `yii\data\SqlDataProvider`
- `ActiveDataProvider` akan berisi data dalam bentuk objek, sedangkan `ArrayDataProvider` dan `SqlDataProvider` akan berisi data dalam bentuk array.

# ActiveDataProvider

- ActiveDataProvider digunakan untuk menyediakan data dengan melakukan query melalui `yii\db\ActiveQuery` atau `yii\db\query` (Perhatikan contoh berikut).

```
// in the View
// ...
$post = $provider->getModels();
foreach ($posts as $post) {
    // display the $post data here
}

// display pagination
echo \yii\widgets\LinkPager::widget([
    'pagination' => $provider->pagination,
]);
// ...
```

```
// in the Controller
// using yii\db\ActiveQuery
function actionIndex()
{
    $provider = new ActiveDataProvider([
        'query' => Post::find(),
        'pagination' => [
            'pageSize' => 10,
        ],
    ]);
    return $this->render('index', [
        'dataProvider' => $provider]);
}
```

```
// in the Controller
// using yii\db\Query
function actionIndex()
{
    $query = new Query();
    $provider = new ActiveDataProvider([
        'query' => $query->from('post'),
        'pagination' => [
            'pageSize' => 10,
        ],
    ]);
    return $this->render('index', [
        'provider' => $provider]);
}
```

# SqlDataProvider

- SqlDataProvider digunakan untuk menyediakan data menggunakan script SQL dasar.
- Seperti Data Provider lainnya, SqlDataProvider juga mendukung fitur Pagination dan Sorting.
- Perhatikan contoh berikut:

```
// in the View  
// ...  
$users = $provider->getModels();  
foreach ($posts as $post) {  
    // display the $post data here  
}  
  
// display pagination  
echo \yii\widgets\LinkPager::widget([  
    'pagination' => $provider->pagination,  
]);  
// ...
```

```
// in the Controller  
function actionIndex()  
{  
    $count = Yii::$app->db->createCommand('  
        SELECT COUNT(*) FROM user')  
        ->queryScalar();  
  
    $provider = new SqlDataProvider([  
        'sql' => 'SELECT * FROM user',  
        'totalCount' => $count,  
        'sort' => [  
            'attributes' => [  
                'age',  
                'name'  
            ],  
        ],  
        'pagination' => [  
            'pageSize' => 10,  
        ],  
    ]);  
  
    return $this->render('index', [  
        'provider' => $provider]);  
}
```

# Data Widgets

- Data Widgets merupakan widgets yang berfungsi untuk menampilkan data.
- Yii menyediakan beberapa Data Widgets, yaitu:
  - DetailView (secara default digunakan pada view "View" yang di-generate menggunakan Gii).
  - ListView
  - GridView (secara default digunakan pada view "Index" yang di-generate menggunakan Gii).

# Latihan

- Gunakan pagination pada halaman depan dari aplikasi toko online anda!
- Tambahkan fitur untuk menampilkan item produk berdasarkan kategori!



**Terima Kasih**