JMS SERIALIZER

```
use JMS\Serializer\Annotation\Type;

class BlogPost
{
    /**
     * @Type("ArrayCollection<MyNamespace\Comment>")
     */
    private $comments;
    /**
     * @Type("string")
     */
    private $title;
    /**
     * @Type("MyNamespace\Author")
     */
    private $author;
    /**
     * @Type("DateTime")
     */
    private $startAt;
    /**
     * @Type("DateTime<'Y-m-d'>")
     */
    private $endAt;
```

ZEND HYDRATOR

Zend\Hydrator\ObjectProperty

Any data key matching a publicly accessible property will be hydrated; any public properties will be used for extraction.

Zend\Hydrator\Reflection

Similar to the `ObjectProperty` hydrator, but uses PHP's reflection API to hydrate or extract properties of any visibility. Any data key matching an existing property will be hydrated; any existing properties will be used for extraction.

Zend\Hydrator\ArraySerializable

Follows the definition of `ArrayObject`. Objects must implement either the `exchangeArray()` or `populate()` methods to support hydration, and the `getArrayCopy()` method to support extraction.

Zend\Hydrator\ClassMethods

Any data key matching a setter method will be called in order to hydrate; any method matching a getter method will be called for extraction.