



Kolejkowanie zapytań AJAX

rozszerzenie xsAjaxQueues dla jQuery

jak...

...upewnić się, że następne zapytania będą wysyłane dopiero po odebraniu wcześniej wysłanych zapytań?

...w prosty sposób decydować o kolejności wysłania zapytań?

...zapobiec wykonaniu funkcji wywoływanej po odebraniu odpowiedzi na zapytanie, jeśli nie uzyskano jeszcze odpowiedzi na inne zapytanie?

jak...

...w prosty sposób kontrolować stan
zapytania i zmieniać funkcje wywoływane
po odebraniu odpowiedzi?

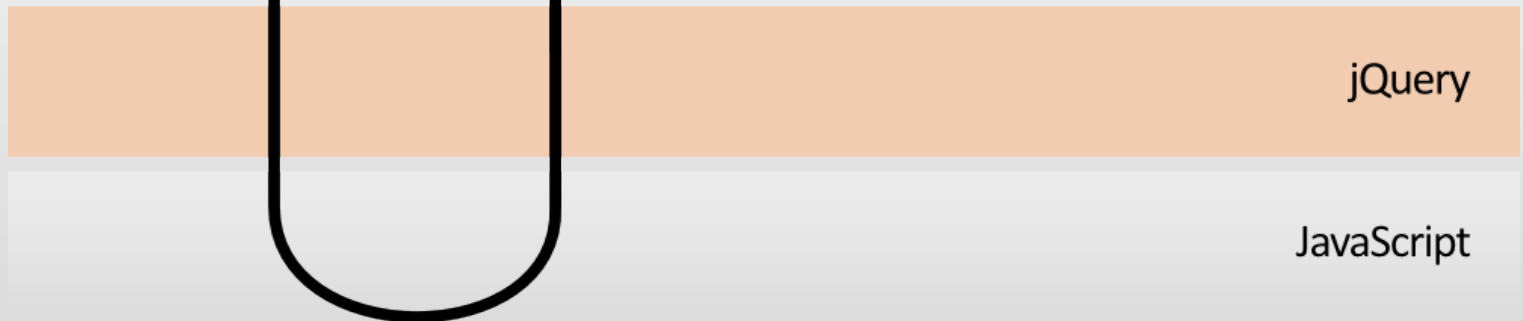
...uzyskać wpływ na obsługę zapytań
wysyłanych przez obcy kod
bez jego modyfikacji?

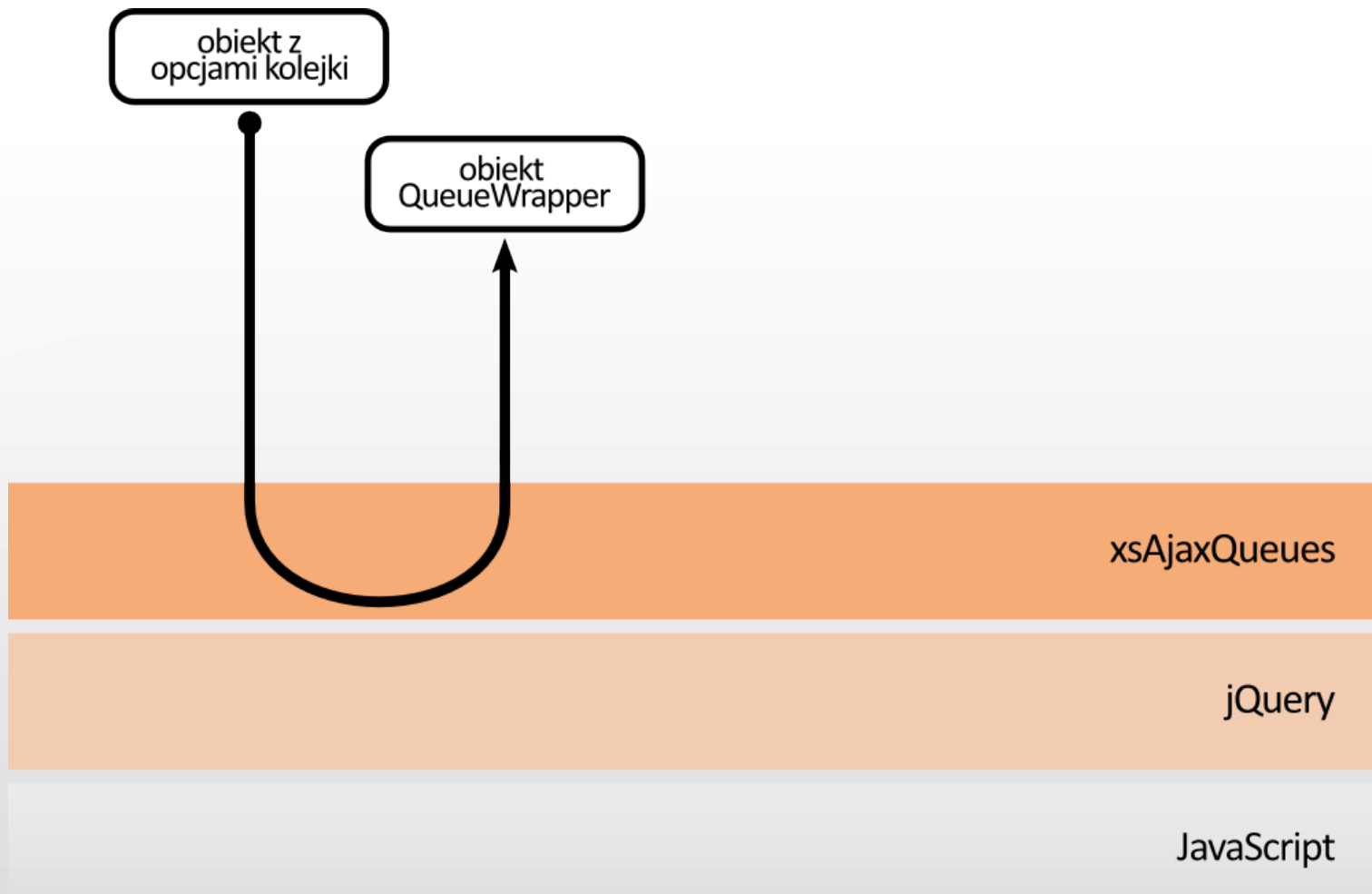
obiekt z
opcjami zapytania

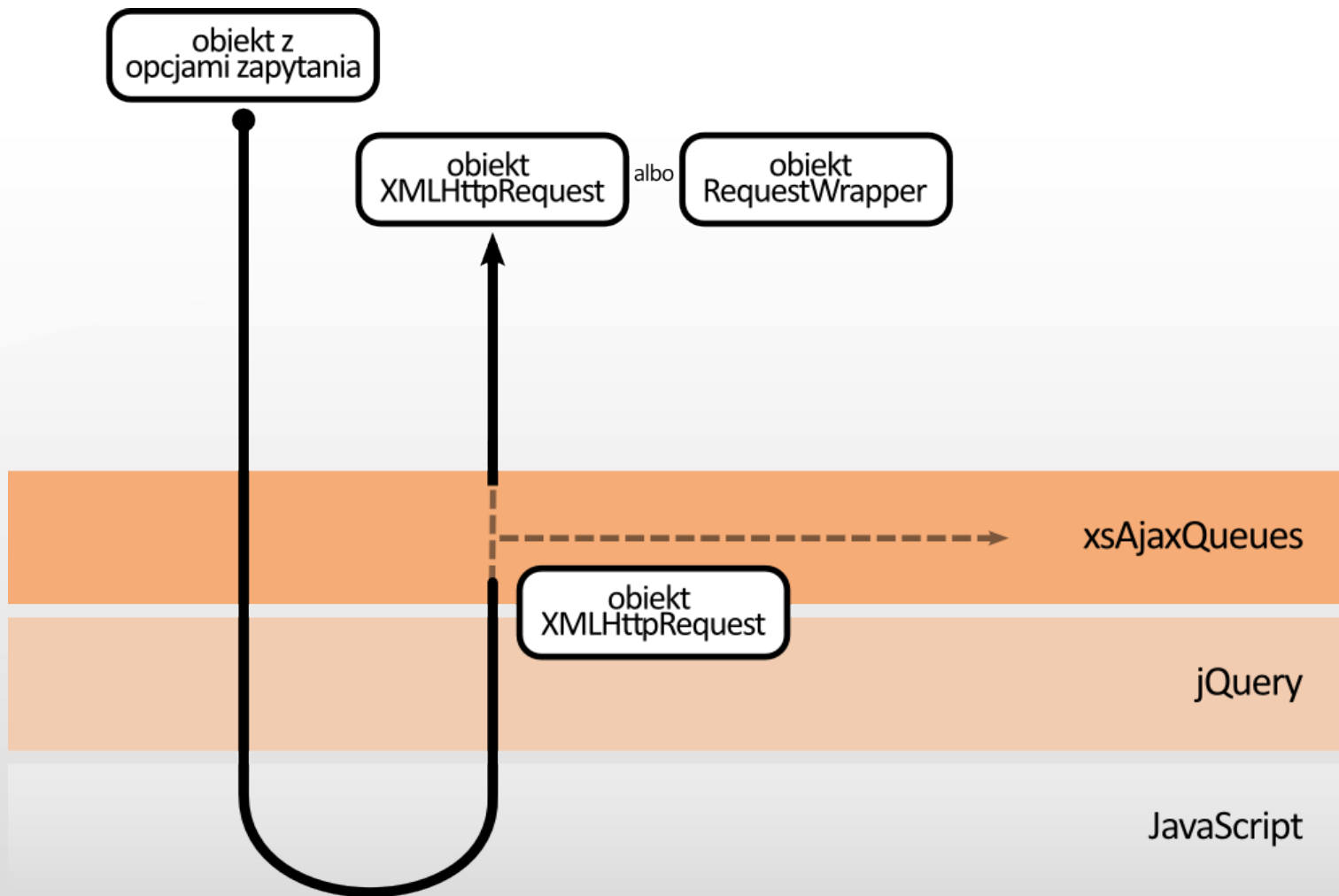
obiekt
XMLHttpRequest

jQuery

JavaScript







umożliwienie korzystania z kolejek:

```
<script type="text/javascript"  
src="jquery-1.4.2.js"></script>
```

```
<script type="text/javascript"  
src="xsAjaxQueues.js"></script>
```

utworzenie kolejki:

```
var queueHandle = $.ajaxQueue({  
    id: 'my_queue',  
    order: 'fifo' | 'lifo',  
    mode: 'request' | 'response',  
    priority: undefined | 'asc' | 'desc'  
});
```

uzyskanie uchwytu do wcześniej utworzonej kolejki:

```
var queueHandle = $.ajaxQueue('my_queue');
```


wysłanie tradycyjnego zapytania:

```
var XMLHttpRequest = $.ajax({  
  type: 'post',  
  data: { ... },  
  ...  
});
```

wysłanie zapytania dołączonego do określonej kolejki:

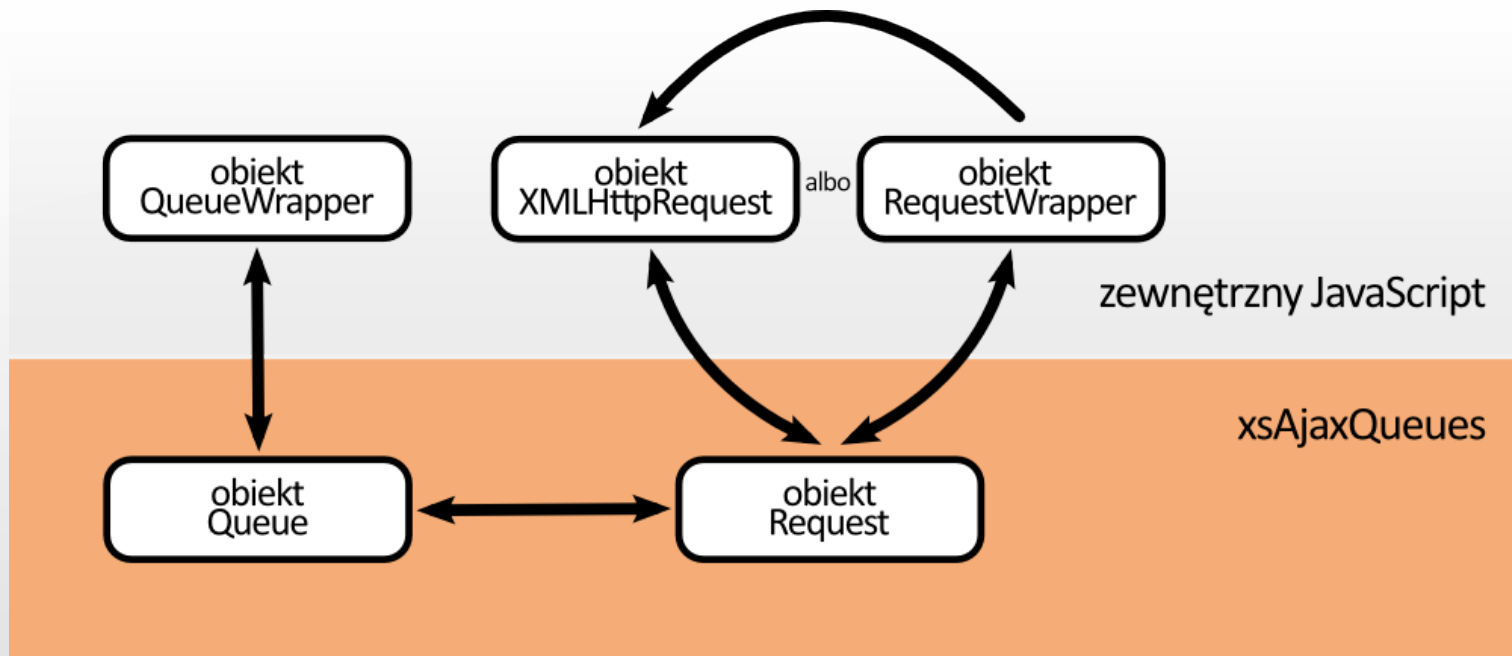
```
var XMLHttpRequest = $.ajax({  
  type: 'post',  
  data: { ... },  
  queue: 'my_queue'  
  ...  
});
```

wysłanie zapytania o określonym priorytecie:

```
var XMLHttpRequest = $.ajax({  
  type: 'post',  
  data: { ... },  
  queue: 'my_queue',  
  priority: 7,  
  ...  
});
```

wysłanie zapytania i pobranie uchwytu do jego obiektu:

```
var requestHandle = $.ajax({  
  type: 'post',  
  data: { ... },  
  queue: 'my_queue',  
  priority: 7,  
  wrapper: true,  
  ...  
});
```

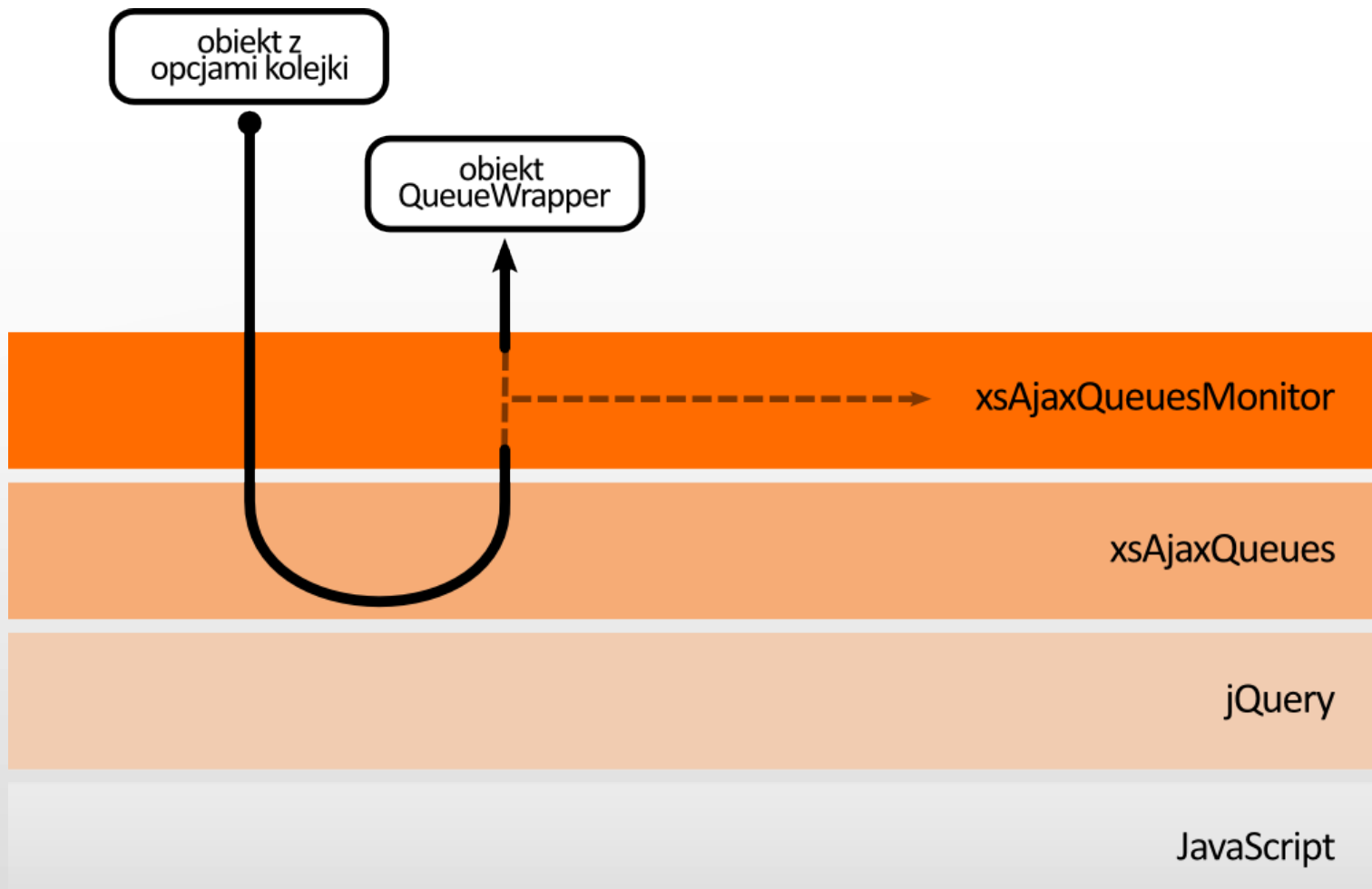


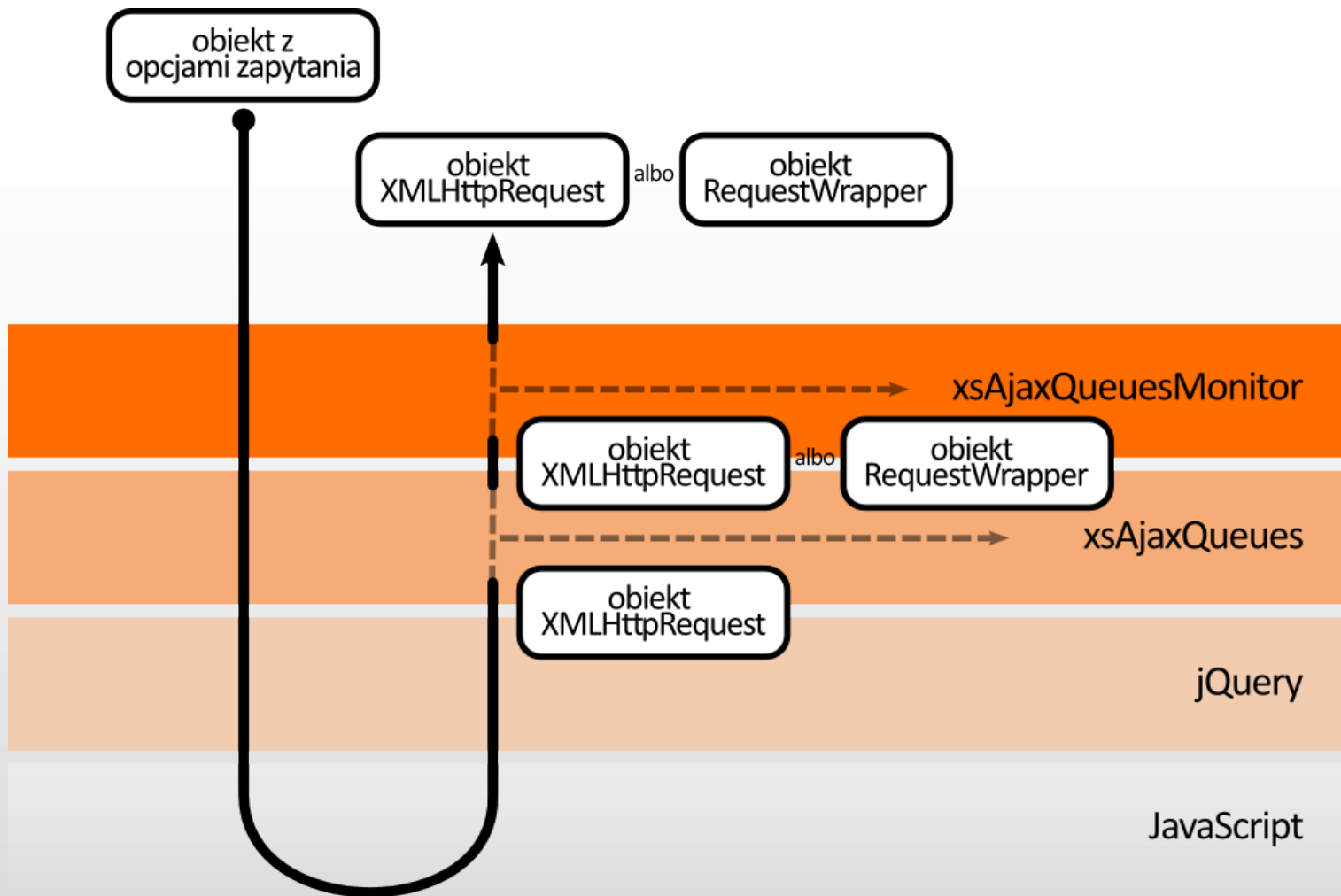
metody uchwytu do obiektu kolejki:

```
.getId()  
.getSettings()  
.isLocked()  
.isEnabled()  
.enable()  
.disable()  
.toggle()  
.cancel()  
.getRequests()  
.getCounters()  
.addListener(listener)  
.removeListener(listener)
```

metody uchwytu do obiektu zapytania:

<code>.getId()</code>	<code>.getXMLHttpRequest()</code>
<code>.getSettings()</code>	<code>.getCallback(name)</code>
<code>.getQueue()</code>	<code>.setCallback(name, callback)</code>
<code>.getQueueId()</code>	<code>.clearCallback(name)</code>
<code>.getFlags()</code>	<code>.cancel(abort)</code>
<code>.isSent()</code>	<code>.getCallbackArgs(name)</code>
<code>.isReceived()</code>	<code>.getCallbackContext(name)</code>
<code>.isProcessed()</code>	<code>.addListener(listener)</code>
<code>.isCanceled()</code>	<code>.removeListener(listener)</code>
<code>.isAborted()</code>	





umożliwienie korzystania z kolejek:

```
<script type="text/javascript"  
src="jquery-1.4.2.js"></script>
```

```
<script type="text/javascript"  
src="jquery-xAjaxQueues.js"></script>
```

```
<script type="text/javascript"  
src="jquery-ui-1.8.2.custom.js"></script>
```

```
<link type="text/css"  
href="ui-lightness/jquery-ui-1.8.2.custom.css"  
rel="stylesheet" />
```


dodanie filtra przechwytyjącego:

```
$.ajaxQueue(function(settings) {  
    if (settings.url === 'some_url.php') {  
        settings.queue = 'intercepted_requests';  
        settings.priority = 5;  
        return settings;  
    }  
});
```

Dziękuję za uwagę!