

Ahmes Sp. z o.o.

ul. Lewicka 13/15

02-547 Warszawa

tel: (22) 113 10 00, fax: (22) 203 63 21, e-mail: biuro@ahmes.pl, <http://www.ahmes.pl>



**DOKUMENTACJA
IMPLEMENTACJI MECHANIZMÓW OBSŁUGI
AHMES SMS (html)
2013-03-04**

Spis treści

WPROWADZENIE.....	3
WYSYŁANIE SMS'ów	4
Proste wysłanie wiadomości	4
Ustalenie typu wysyłanej wiadomości	4
Ustawienie daty wysłania wiadomości.....	5
Ustawienie grupy kontaktów, jako odbiorcę.....	5
KSIĄŻKA ADRESOWA	6
Wylistowanie grup w książce adresowej.....	6
Dodanie grupy do książki adresowej	6
Usunięcie grupy z książki adresowej	7
Wylistowanie numerów danej grupy w książce adresowej	7
Dodawanie numeru do danej grupy książki adresowej.....	8
Usuwanie numeru z grupy książki adresowej	8
HISTORIA SMS'ów	9
Wylistowanie wysłanych wiadomości	9
Stronicowanie wysłanych wiadomości	9
Filtrowanie po dacie wysłanych wiadomości	9
Pobieranie historii w formacie CSV	10
ZAPLANOWANE WIADOMOŚCI SMS.....	10
Wylistowanie zaplanowanych wiadomości SMS.....	10
Usuwanie zaplanowany wiadomości.....	11

WPROWADZENIE

Dokumentacja pokazuje dostęp do usługi przy pomocy biblioteki napisanej w języku JavaScript. Rozwiązanie pozwala pisać interfejs do usługi tylko za pomocą języka HTML5. Do korzystania z usługi potrzebne są parametry *user_name* i *user_password* odzwierciedlające dane uwierzytelniające w witrynie <http://partners.ahmes.pl>.

Na każdej stronie web, która będzie korzystać z naszej biblioteki, dodać należy odwołanie do trzech bibliotek:

```
<script src="http://code.jquery.com/jquery-1.9.1.js">
</script>
<script type="text/javascript" src="http://knockoutjs.com/downloads/knockout-2.2.1.js">
</script>
<script type="text/javascript" src="http://partners.ahmes.pl/api/ahmessmsapi-1.0.js">
</script>
```

WYSYŁANIE SMS'ów

Do wysyłania sms'ów służy funkcja wewnętrzna o nazwie `SendSmsViewModel`

Przykład zaimplementowania funkcji na stronie:

```
<script type="text/javascript">
    $(document).ready(function () {
        ko.applyBindings(new SendSmsViewModel("[nazwa użytkownika]", "[hasło]"));
    });
</script>
```

Proste wysłanie wiadomości

Do wysłania w prosty sposób wiadomości należy użyć np. dwóch znaczników `input`. Jednego do wprowadzenia treści wiadomości i jednego do wprowadzenia numeru adresata. Potrzebny jest także element do wywołania akcji wysłania wiadomości np. `button`.

Przykład użycia:

```
<input data-bind="value: To" />
<textarea data-bind="value: Message"></textarea>
<button data-bind="click: sendSms">Wyślij wiadomość</button>
```

Jeżeli wiadomość zostanie pozytywnie wysłana pola(`To` i `Message`) zostaną wyczyszczone.

Ustalenie typu wysyłanej wiadomości

Rozróżnia się dwa typy wiadomości do wysłania(SMS i FLASH). Domyślnie wysyłane są wiadomości typu SMS. Dla zmiany typu wiadomości należy użyć np. znaczników `input` typu `radio`.

Przykład użycia:

```
<label>
    <input
        type="radio"
        checked="checked"
        name="MessageType" value="1"
        data-bind="checked: SendSmsType"/>
    SMS
</label>
<label>
    <input
        type="radio"
        name="MessageType"
        value="0"
        data-bind="checked: SendSmsType"
        style="width:25px;"/>
    FLASH
</label>
```

Ustawienie daty wysłania wiadomości

Usługa pozwala ustawić datę wysłania wiadomości. Jeżeli data zostanie ustawiona na wcześniejszą, lub równą bieżącej to wiadomość zostanie wysłana natychmiast. Datę należy wstawiać np. do znaczników *input* typu *text* w formacie 'RRRR-MM-DD GG:MM:SS'.

Przykład użycia:

```
<input type="text" data-bind="value: SendSmsDate" />
```

Ustawienie grupy kontaktów, jako odbiorcę

Jedną z funkcjonalności jest wybór odbiorców wiadomości między indywidualnym numerem, a grupą kontaktów (Edycja grupy kontaktów opisana jest poniżej). Należy użyć wtedy np. standardowego znacznika *input* typu *text* do wprowadzania indywidualnego odbiorcy, oraz znacznika *select* do wprowadzania grupy odbiorców. Wybór pomiędzy typami odbiorców można zaimplementować przy użyciu znaczników *input* typu *radio*.

Przykład użycia:

```
<input data-bind="value: To, visible: SendToType() === '1'" />
<select
  data-bind="options: AddressBookCollection, value: GroupName, visible: SendToType() ===
    '0', optionsCaption: 'Wybierz grupę...'" />

<h3>Typ odbiorcy:</h3>
<label>
  <input type="radio" checked="checked" value="1" name="ReceiverType" data-
    bind="checked:SendToType, valueUpdate: 'keydown'" style="width:25px;"/>
  Numer
</label>
<label>
  <input type="radio" name="ReceiverType" value="0" data-bind="checked:SendToType,
    valueUpdate: 'keydown'" style="width:25px;"/>
  Grupa
</label>
```

KSIĄŻKA ADRESOWA

Do edycji książki adresowej służy funkcja wewnętrzna o nazwie `AddressBookViewModel`.

Przykład zaimplementowania funkcji na stronie:

```
<script type="text/javascript">
  $(document).ready(function () {
    ko.applyBindings(new AddressBookViewModel("[nazwa użytkownika]", "[hasło]"));
  });
</script>
```

Wylistowanie grup w książce adresowej

Do wylistowania grup książki adresowej należy użyć tabeli, lub znaczników do listujących (*ul*).

Przykład użycia:

```
<table>
  <thead>
    <tr>
      <th>Nazwa grupy</th>
      <th>Data dodania</th>
    </tr>
  </thead>
  <tbody data-bind="foreach: AddressBookCollection">
    <tr>
      <td data-bind="text: Name"></td>
      <td data-bind="text: CreateDate"></td>
    </tr>
  </tbody>
</table>
```

Dodanie grupy do książki adresowej

Aby dodać grupę kontaktów należy wprowadzić nazwę grupy np. przy użyciu znacznika *input* typu *text*, oraz wywołać zdarzenie dodania np. w formularzu przy użyciu znacznika *button*.

Przykład użycia:

```
<form data-bind="submit: addAddressBookClick">
  <input data-bind="value: AddressBookNameToAdd"/>
  <button type="submit" />Dodaj grupę</button>
</form>
```

W przypadku poprawnego dodania grupy pole nazwa grupy w polu *input* zostanie usunięta i zostanie dodana nowa nazwa grupy do listy grup, w przeciwnym wypadku zostanie wywołany w przeglądarce komunikat o błędzie w dodaniu grupy.

Usunięcie grupy z książki adresowej

Aby usunąć grupę z książki adresowej należy np. dodać nową kolumnę w tabeli listującej wszystkie grupy i dodać w niej znacznik *a* usuwający daną grupę.

Przykład użycia:

```
<tr>
  <td data-bind="text: Name"></td>
  <td data-bind="text: CreateDate"></td>
  <td><a href="#" data-bind="click: $root.removeAddressBookClick">Usuń</a></td>
</tr>
```

W przypadku poprawnego usunięcia grupy zostanie ona usunięta z listy grup, w przeciwnym wypadku zostanie wywołany w przeglądarce komunikat o błędzie w usunięciu grupy.

Wylistowanie numerów danej grupy w książce adresowej

Żeby wylistować wszystkie numery danej grupy należy ustawić daną grupę na liście, jako zaznaczoną. Jest to możliwe poprzez dodanie dodatkowego parametru, do danego elementu na liście.

Przykład użycia:

```
<tr>
  <td data-bind="text: Name, click: $root.selectAddressBookClick"></td>
  <td data-bind="text: CreateDate, click: $root.selectAddressBookClick"></td>
</tr>
```

Zaznaczoną grupę kontaktu można wyświetlić np. w formie tabeli, albo listy (*ul*).

Przykład użycia:

```
<table>
  <thead>
    <tr>
      <th>Numer</th>
      <th>Data dodania</th>
    </tr>
  </thead>
  <tbody data-bind="foreach: AddressBookNumberCollection">
    <tr>
      <td data-bind="text: Number"></td>
      <td data-bind="text: CreateDate"></td>
    </tr>
  </tbody>
</table>
```

Dodawanie numeru do danej grupy książki adresowej

Aby dodać numer do grupy książki adresowej należy najpierw zaznaczyć daną grupę i wprowadzić nazwę grupy np. przy użyciu znacznika *input* typu *text*, oraz wywołać zdarzenie dodania np. w formularzu przy użyciu znacznika *button*.

Przykład użycia:

```
<form data-bind="submit: addAddressBookNumberClick">
  <input data-bind="value: AddressBookNumberMsisdnToAdd"/>
  <button type="submit" />Dodaj numer</button>
</form>
```

W przypadku poprawnego dodania numeru do grupy numer dodany w polu *input* zostanie wyczyszczone i zostanie dodany numer do widocznej grupy, w przeciwnym wypadku zostanie wywołany w przeglądarce komunikat o błędzie w dodaniu grupy.

Usuwanie numeru z grupy książki adresowej

Aby usunąć numer z grupy należy np. dodać nową kolumnę w tabeli listującej wszystkie numery i dodać w niej znacznik *a* usuwający dany numer.

Przykład użycia:

```
<tr>
  <td data-bind="text: Number"></td>
  <td data-bind="text: CreateDate"></td>
  <td>
    <a href="#" data-bind="click: $root.removeAddressBookNumberClick">Usuń</a>
  </td>
</tr>
```

W przypadku poprawnego usunięcia numeru zostanie on usunięty z listy, w przeciwnym wypadku zostanie wywołany w przeglądarce komunikat o błędzie w usunięciu numeru.

HISTORIA SMS'ów

Do przeglądania historii wysłanych smów służy funkcja wewnętrzna o nazwie `HistoryViewModel`.

Przykład zaimplementowania funkcji na stronie:

```
<script type="text/javascript">
  $(document).ready(function () {
    ko.applyBindings(new HistoryViewModel("[nazwa użytkownika]", "[hasło]"));
  });
</script>
```

Wylistowanie wysłanych wiadomości

Do wylistowania historii wysłanych wiadomości należy użyć tabeli, lub znaczników listujących (`ul`).

Przykład użycia:

```
<tbody data-bind="foreach: SendSmsCollection">
  <tr>
    <td data-bind="text: To"></td>
    <td data-bind="text: Report"></td>
    <td data-bind="text: SendDate"></td>
  </tr>
</tbody>
```

Stronicowanie wysłanych wiadomości

Do stronicowania historii wysłanych wiadomości należy ustawić ile będzie elementów na danej stronie np. przy użyciu znacznika *select* i uruchomić zdarzenie przejścia do następnej/poprzedniej strony np. za pomocą znacznika *button*.

Przykład użycia:

```
<button data-bind="click: PrevPageClick" >Poprzednia</button>
<strong data-bind="text: PageSelected"></strong>
<button data-bind="click: NextPageClick" >Następna</button>
<select data-bind=" options: AvailablePageCounts, value: PageCount, event: { change:
PageCountChanged }"></select>
```

Filtrowanie po dacie wysłanych wiadomości

Do filtrowania historii wysłanych wiadomości po dacie należy wstawić np. dwa pola *input* (data początkowa filtru i końcowa) typu *text*. Data edytowana w tych polach(np. przy użyciu pluginów) powinna być w formacie 'RRRR-MM-DD GG:MM:SS'.

Przykład użycia:

```
<input data-bind="value: StartDate, event: { change: PageCountChanged }" />
<input data-bind="value: EndDate, event: { change: PageCountChanged }" />
```

Pobieranie historii w formacie CSV

Do pobierania historii wiadomości należy wywołać zdarzenie np. przy użyciu znacznika *button*. Pobierane wyniki mogą być filtrowane po dacie tak jak to zostało opisane w podpunkcie powyżej.

Przykład użycia:

```
<button data-bind="click: DownloadCsvCollectionClick">Pobierz w pliku CSV</button>
```

ZAPLANOWANE WIADOMOŚCI SMS

Do przeglądania i usuwania zaplanowanych wiadomości służy wewnętrzna funkcja *PlannedSmsViewModel*.

Przykład zaimplementowania funkcji na stronie:

```
<script type="text/javascript">
$(document).ready(function () {
    ko.applyBindings(new PlannedSmsViewModel("[nazwa użytkownika]", "[hasło]"));
});
</script>
```

Wylistowanie zaplanowanych wiadomości SMS

Do wylistowania zaplanowanych wiadomości należy użyć tabeli, lub znaczników listujących (*ul*).

Przykład użycia:

```
<table>
  <thead>
    <tr>
      <th>Adresat</th>
      <th>Data wysłania</th>
      <th>ID grupy odbiorców</th>
    </tr>
  </thead>
  <tbody data-bind="foreach:PlannedSmsCollection">
    <tr>
      <td data-bind="text: To"></td>
      <td data-bind="text: SendDate"></td>
      <td data-bind="text: MessageId"></td>
    </tr>
  </tbody>
</table>
```

Usuwanie zaplanowanych wiadomości

Aby usunąć pojedynczego odbiorcę, lub wszystkich odbiorców z zaplanowanej wiadomości należy np. dodać dodatkowe kolumny do tabeli listującej zaplanowane wiadomości wywołujące dane zdarzenia.

Przykład użycia z dwoma metodami usuwania:

```
<tr>
  <td data-bind="text: To"></td>
  <td data-bind="text: SendDate"></td>
  <td data-bind="text: MessageId"></td>
  <td>
    <a href="#" data-bind="click: $root.DeletePlannetSmsSingleClick">
      Usuń odbiorcę
    </a>
  </td>
  <td>
    <a href="#" data-bind="click: $root.DeletePlannetSmsGroupClick">
      Usuń grupę odbiorców
    </a>
  </td>
</tr>
```

W przypadku poprawnego usunięcia odbiorcy zostanie usunięty z listy, w przeciwnym wypadku zostanie wywołany w przeglądarce komunikat o błędzie w usunięciu odbiorcy.