

Portal Engine - HAPI

Monday, January 4, 2016 9:43 AM

The portal engine can convert HTML snippets into API calls to dynamically load data, essentially giving you server side programming with straight up HTML and PeopleVine data or sombrero (`{@@}`) tags. Below are common uses of this technology:

Leverage these HTML tags inline to a DIV, span, etc.:

- **Outer Block** - this is the wrapper for the element to be displayed.
 - **data-pv-component** - this is used to identify which PeopleVine component you are loading data for
 - **data-pv-type** - this signifies the type of content to load on the screen, typically a repeater or form.
 - **data-pv-id** - used to load a specific resource on the page e.g. a specific product, event or content snippet. Otherwise it will pull from the query string parameters.
 - **data-pv-parameters** - you can place a JSON parsed string in here to query through the data (see API)
 - You can also access referenceTypes passed in by using `{@referenceType.survey@}` which will bring back the survey_no of the submission
 - **data-pv-object** - will automatically place the data object in a `<script>` tag for javascript purposes
 - **data-pv-conditional** - will only render this code if the condition is met (e.g. `data-pv-conditional="{@sport@}=Baseball"` where `{@sport@}` is the parameter and Baseball is the value)
 - Supports **attribute(NAME OF ATTRIBUTE)** instead of
 - `{@FIELD@}` pulls from the parameters or querystring
 - Now works with `data-pv-do="conditional"`
 - **data-pv-noresultsmessage**="your message here" will display this message if no data is returned
 - **data-pv-noresultshide="id or class"** and we will hide it if there are no results by adding a `display:none` in the css on the page
 - **data-pv-noresultsshow="id or class"** and we will hide it if there are no results by adding a `display:none` in the css on the page
 - **data-pv-skip** = will skip the rendering
 - **data-pv-report="true"** - will auto generate the report in a pageable data table (only works for general:report)
 - `<script type="text/javascript">var table = $(''.autoDataTable'); table.DataTable({ responsive: true });</script>`
- **Inner Block** - this is the repeating template that displays the content
 - **data-pv-element** - this is the data field to be loaded in between the HTML tags. These can be identified within the API documentation
 - Enter the element name (e.g. `first_name`)
 - Get a secondary element with `.` (e.g. `customer.first_name`)
 - Get a value from array with `[]` (e.g. `social_networks[facebook]`)
 - **data-pv-formatter** - this can be used in combination with a `data-pv-element` to format the data to the end user.
 - **data-pv-min** & **data-pv-max** - this is used to set the min and max on a select dropdown
 - **data-pv-link** - Grabs the url associated with the tag and sets it as the href
 - **data-pv-firstclass** - used to assign this class to the first element
 - **data-pv-prefix** - this will prefix the value with this text
 - **data-pv-suffix** - this will suffix the value with this text
 - **data-pv-repeater** - this allows you to use a subset of data and parse through it (e.g. responses in `surveyResponseData`)
- **Forms**
 - `<input type="button" onclick="HAPIform(#ID, ref_type)"/>` - this will submit all the input and select options to our HAPIform method which will then submit the contents to the `manageltems` method to create the item

These are the type of components and types that can be leveraged:

- Events & Ticketing (**data-pv-component="event"**)
 - **eventList** - this will display a list of upcoming events based on the query string and/or `data-pv-parameters`.

- **eventView** - this will display the event details
 - **Attendees** - brings back attendee ticket info
- **Content (data-pv-component="content")**
 - **snippet** - this will display a content snippet in the specified HTML tag
 - **list** - this will display a list of content pages based on parameters
 - **page** - this brings back a single page
 - **rotators** - this will return the rotators for a specific group
 - **Faq** - will return FAQ data
- **Contest (data-pv-component="contest")**
 - **leaderboard** - this would display the current results of the contest including all contestants
- **Campaign (data-pv-component="campaign")**
 - **list** - this displays a list of items in a campaign that have been completed and the next one in play
- **Scheduler (data-pv-component="scheduler")**
 - **Item** - this provides the full details on a scheduler
 - **days** - this provides a list of the available dates using start_date and slots_available
 - **Days-available** - does same as days but for reservations
 - **slots** - this provides a list of available schedule slots
 - **Bookings** - this returns all the bookings for a customer. Customer_no must be supplied
 - **Booking** - this will return a single booking either by schedule_book_no, schedule_slot_no or schedule_item_no along with customer_no
- **About (data-pv-component="about")**
 - **socialLink** - then uses data-pv-link tag to load the correct URL
- **Company (data-pv-component="company")**
 - **<empty>** - this displays the company information inline
- **Directory (data-pv-component="directory")**
 - **business_locations** - this lists out the business and business locations
 - **business** - this will return a business or businesses matching criteria
 - **Businessfilters** - returns filters to setup directory filters
- **Media (data-pv-component="media")**
 - **<empty>** - this outputs all media based on the search
- **Survey (data-pv-component="survey")**
 - **info** - returns basic info about the survey
 - **questions** - writes out all the questions/form fields (use {@input@} for the input field)
 - **response** - returns the responses someone added for a survey
 - **responses** - returns all the responses for a survey group, survey, customer or business
 - **formresponses** - returns all responses for a survey and allows to filter by status
 - **application** - returns survey application
 - **application_responses** - returns responses to an application
 - **applicationStats** - return the stats for an application (survey/group)
- **Categories (data-pv-component="categories")**
 - **<empty>** - allows you to search for categories
 - Return Gallery categories: component_no=-1
- **Request (data-pv-component="request")** - this allows you to call an external URL and convert the data to an object
 - **<enter URL>** - make sure you set data-pv-object="obj" to set the data to the javascript for processing
- **Memberships (data-pv-component="membership")**
 - **List** - returns a list of membership types and filterable
 - **Info** - returns 1 that matches the membership_no
 - **Members** - returns a list of members
 - **Memberships** - brings back person's memberships
 - **Activity** - provides recent activity on a member
 - **ActivityStats** - provides stats on recent activity for members
 - **Membershipcard** - returns the membership card details
- **Text (data-pv-component="text")**
 - This will just output the text or HTML that's in the object; it's used for conditional content.
- **Gift Cards (data-pv-component="giftcard")**
 - **<empty>** - This will output the gift card and messaging associated with it.

- Stats - this will return general stats about balance
- Notifications (**data-pv-component="notification"**)
 - **on-page** - this will return on page notifications based on settings
- Bookings (**data-pv-component="bookings"**)
 - Returns all bookings of events and scheduler
- Loyalty (**data-pv-component="loyalty"**)
 - Earned - returns a point_earned activity
 - Balance - returns point balance
 - Activities - returns all activities they can perform
 - Dashboard - returns total points, points until next reward, next level, total visits, total rewards available
- Commerce (**data-pv-component="commerce"**)
 - Shop - searches products as if it were on the /shop page
 - **Mini** - same as shop but smaller data set (productMini)
 - CategoryStats - displays the category statistics
 - Orders - list all order data for person
 - **Locationrevenue** - returns member vs. total spend per location
- Attributes (**data-pv-component="attribute"**)
- Customer (**data-pv-component="customer"**)
 - Attributes - returns all attributes for a customer (good for data-pv-object)
 - Profile - returns the standard customer fields
 - Touchpoints - returns the full customer profile with touchpoints
 - **Demographicdata** - returns general stats about the customer's database with age and gender
 - You can also get to customer attributes by adding **{@customer.attribute(attribute_name|default)@}** anywhere in page
 - **Attributedata** - returns customer attribute group data
- Transaction (**data-pv-component="transaction"**)
 - <blank> returns list of transactions
 - **Stats** - returns general stats for display
 - **Spend** - returns a report of customer spend
 - **Paymentinfo** - returns all payment info on file for member
- Messaging (**data-pv-component="messaging"**)
 - Messages - returns all messages in list
 - Conversation - returns conversations
- Chat (**data-pv-component="chat"**)
 - Thread - returns a single thread
 - Conversation - returns all conversations
- Favorite (**data-pv-component="favorite"**)
- Offers (**data-pv-component="offers"**)
 - List - provides a list of offers based on criteria
 - Filters - returns a list of available filters
- Subscription (**data-pv-component="subscription"**)
 - <blank> returns list of subscriptions
 - **Stats** - returns general stats for display
- Funding Campaigns (**data-pv-component="fundcampaign"**)
 - Campaign - provides the campaign with last 5 posts and donations (can be random)
 - Posts - returns posts by the publisher
 - Donations - returns donations made on the campaign
- Projects (data-pv-component="project")
 - List - returns all projects
 - Bids - returns all bids
 - Deliverables - returns all deliverables for project
- Notes (data-pv-component="notes")
- **General** (data-pv-component="general")
 - **Report** - returns all custom data export reports
 - **Data** - returns the data of a report in JSON format
- **Data** (data-pv-component="data")

- Year - returns the next 25 years in {@val@}
- States - returns a list of states {@key@} {@value@}
- Countries - returns a list countries {@key@} {@value@}

Here's a list of available formatters:

- date() - this is used to format a date to a specific format
 - date(string) - e.g. MMMM MM dd yyyy h:mm tt - allows you to format to a specific date string (based on the .NET string.format rules) If you put in ## we will output st, nd, rd or th based on the day
 - date(month) - display the full month
 - date(day) - display the day number
 - date(year) - display the 4 digit year
 - date(time) - displays the time
 - Date(full) - Full Date with Time
- Ago - formats as duration and adds ago
- Currency - this is used to format to USD
- Phone - this is used to format as phone number
- Number - this will format as number with comma
- Ifblank - will remove this section if blank
- Url - will format as URL (add http if doesn't exist)
- Trim(#) - will trim down to number of characters and add ... to the end
- Percent - returns as a %
- Percentround - returns as a % rounded to whole number
- hasValue - this will return True if there's a value or False if empty
- thumb - returns a thumbnail preview of media object
- Filename - it will parse the filename from a longer URL
- Left - gets left(num of chars)
- Right - gets right(num of chars)
- Subtring - gets substring(start, num of chars)
- Replace - returns replace(find, with)
- Age - returns age of date
- Gender - returns long name for gender
- Replace(x,y) - replace value x in y
- Count - gives you total count of records
 - Use #count to get the count of records in a list e.g. {@slot_fee/availability#count:currency@}
- Uriencode - to encode for URL
- Ifthiselse - used to return values if true or false {@field:ifthiselse(val, true, false)@}
- Boolean - used to return true or false if "Y" or "N"
- If - if the value is equal to the 1st slot, then return slot 2, else return blank {@fieldname:if(active,true)@}
- Serialize - {@attributes:serialize@} will output as a JSON object
- Md5 - turn into lowercase md5 hash

Functions in Forms

- {@data(entity, display text, display value, {parameters in json array})@}
 - {@data(location, location_name, #business_location_no#, {})@}
 - When adding # before and after the value field, it will use the raw value, otherwise it will return titel~value~{type}

Here's a list of functions that are supported:

- + to add two values
- - to subtract two values
- / to divide two values
- * to multiply two values
- Cool technique to nest variable and math
 - {@{@q:page_number@}+1@}

Here's a list of parameter variables you can use:

- {@today@} will be replaced with today's date
- {@todaystart@} will be replaced with today's date and 12:00 AM
- {@todayend@} will be replaced with today's date and 11:59 PM
- {@todayjust@} will be replaced with today's date only
- {@yesterday@} will be replaced with yesterday's date
- {@yesterdaystart@} will be replaced with yesterday's date and 12:00 AM
- {@yesterdayend@} will be replaced with yesterday's date and 11:59 PM
- {@yesterdayjust@} will be replaced with today's date only
- {@tomorrow@} will be replaced with tomorrow's date
- {@tomorrowstart@} will be replaced with tomorrow's date and 12:00 AM
- {@tomorrowend@} will be replaced with tomorrow's date and 11:59 PM
- {@tomorrowjust@} will be replaced with tomorrow's date only
- {@today:addDays(#}@} will be replaced with today plus days
- {@today:addMonths(#}@} will be replaced with today plus months
- {@today:addYears(#}@} will be replaced with today plus years
- {@thisWeekStart@} will be first day of this week
- {@thisWeekEnd@} will be last day of this week
- {@lastWeekStart@} will be first day of last week
- {@lastWeekEnd@} will be last day of last week
- {@nextWeekStart@} will be first day of next week
- {@nextWeekEnd@} will be last day of next week
- {@thisMonthStart@} will be first day of this month
- {@thisMonthEnd@} will be last day of this month
- {@nextMonthStart@} will be first day of next month
- {@nextMonthEnd@} will be last day of next month
- {@lastMonthStart@} will be first day of last month
- {@lastMonthEnd@} will be last day of last month
- {@thisMonthName@} will be this month's name
- {@lastMonthName@} will be last month's name
- {@thisMonthNumber@} will be this month's number
- {@lastMonthNumber@} will be last month's number
- {@thisYear@} will be this year
- {@lastYear@} will be last year
- {@thisMonth@} will be this month's name
- {@lastMonth@} will be last month's name
- {@customer_no@} will be replaced with the current active customer_no
- {@reference_no@} will be replaced with the reference_no passed in (mainly for newsletters)
- {@reference_no:event@} where event = reference_type will be replaced with the event's reference_no. This is used for newsletter processing that has multiple referenceTypes such as event and event_ticket
- {@q:field_name@} will replace values with data in query string or parameters included
 - You can combine q:field_name:xyz where xyz is fallback value
 - (e.g. {@q:from_date:1/1/1900@})
- {@var_customer_no@} (only used in content)
- {@var_business_no@} (only used in content)
- {@increment@} is used in repeaters to know which iteration you're on
- {@default_domain@} or {@domain_name@} will put in your default domain
- {@ip_address@}
- {@user_latitude@} and {@user_longitude@} for location based queries
- {@obj@} outputs the object as a serialized string replacing double quotes with quotes
- {@currency_symbol@} returns the locale currency symbol

We will also auto pass in the customer_no and business_no of the logged in user in order to filter the results. If you want to omit the auto populating then add business_no: 0 or customer_no: 0 in the data-pv-parameters list

Conditional Logic supports the following:

- Attribute(name) = X
- Customer.attribute(name) = x
- Customer.total* > or = 0 for any total value in the CRM profile
- Membership(membership_sku) - this does not need a conditional operator, it will trigger if they have access
 - Or leave out membership_sku and will check for any membership
- Subscription(service_sku) - this does not need a conditional operator, it will trigger if they are subscribed
 - Or leave out service_sku and will check for any service subscription
- totalCategories(reference_type) > 0
- companySetting(name) = X
- Permissions(events-create) will check permissions for the business
- Conditional can include:
 - =
 - != or <> for not equals
 - >
 - <
 - >=
 - <=
 - startsWith
 - endsWith
 - contains
- You can concatenate with && for (and) and || for (or).

TIPS

Follow these tips for best practices

- If you're trying to access a value in an array (in the object) then simply use the notation such as element.field and we will supply the field value for the first item in the array.

Data-pv-object

Samples

```
<div data-pv-do="conditional" data-pv-conditional="membership(vault-concierge)">
```

```

  <div data-pv-do="conditional" data-pv-conditional="companySetting(allowSelfFundraiser)=True">
    <a href="/account/fundraiser/create" class="btn btn_hasIcon">Launch a Fundraiser
      <svg class="icon">
        <use xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="/media/svg/defs.svg#icon_plusEnclosedFilled"></use></svg>
      </a>
    </div>
  </div>

```

You can check to see how many categories exist in a section in order to show or not

```
<section class="component component_loose" data-pv-component="text" data-pv-conditional="totalCategories(fundcampaign) > 0">
```

You can add a formatter to the end

```
{@totalRaised/funding_goal:percent@}
```

You can do multiple items even without variables

```
{@totalRaised/funding_goal*100@}
```

You can use attributes to personalize emails and content

```
<div data-pv-component="text" data-pv-conditional="attribute(purchase) = Next 6 Months">
  Next 6 months
```

</div>

You Earned <strong data-pv-component="text" data-pv-conditional="{@reward_type@} = points">6 points<strong data-pv-component="text" data-pv-conditional="{@reward_type@} = offer">an offer for Offer Name for completing Activity Title on DATE.

Handle multiple conditions

```
<span class="pv-text-primary">
  <span data-pv-component="text" data-pv-conditional="{@totalDonations@}>0">
    <i class="fa fa-heart-o"></i>&nbsp;{@totalDonations@} Supporters
  </span>
  <span data-pv-component="text" data-pv-conditional="{@totalDonations@}=0">
    <i class="fa fa-heart-o"></i>&nbsp;Be The First to Support
  </span>
</span>
```

EVENT

```
<h4 data-pv-element="event_title" class="eventTitle">Event Name</h4>
<a href="/event/{@event_no@}">LINK Name</a>
```

```
<ul class="blocks blocks_equalHeight blocks_3to2to1">
  <li data-pv-component="event" data-pv-type="eventList">
    <div class="blockTile blockTile_alignLeft">
      <div class="blockTile-hd blockTile-hd_flush">
        <div class="imgMask imgMask_fluid">
          <a class="" href="/event/{@event_no@}"><img class="imgMask-img" data-pv-alt="event_title" data-
pv-element="event_graphic"></a>
        </div>
      </div>
      <div class="blockTile-bd">
        <ul class="vr vr_x2">
          <li><span class="txt txt_enlarged mix-txt_medium" data-pv-element="event_date" data-pv-
formatter="date(MMMM dd yyyy hh:mm)"></span></li>
          <li>
            <a class="" href="/event/{@event_no@}"><span class="hdg hdg_4" data-pv-element="event_title">
Event Title</span></a>
          </li>
          <li class="vr-override_x6"><span class="txt txt_enlarged" data-pv-element="event_summary">
Summary of the event</span></li>
          <li><span class="hdg hdg_6" data-pv-label="label.venue">Venue</span> <span data-pv-
element="event_venue">Event Venue</span></li>
          <li><span class="hdg hdg_6" data-pv-label="label.location">Location</span> <span data-pv-
element="event_location">Event Location</span> <!-- <a href="#" class="mapsButton" target="_blank" data-pv-
element="event_location"></a--></li>
        </ul>
      </div>
      <div class="blockTile-ft">
        <a class="btn mix-btn_stretch" href="/event/{@event_no@}">View Event</a>
      </div>
    </div>
  </li>
</ul>
```

CONTEST LEADERBOARD

```
<div data-pv-component="contest" data-pv-type="leaderboard" data-pv-parameters="{customer_no:
{@customer_no@}, rank_no: 38}">
  <div data-pv-element="username" class="contestantName">Andrew Knies</div>
  <div data-pv-element="totalPoints" data-pv-formatter="currency" class="contestantPoints">$1,000,000</div>
</div>
```

CONTENT SNIPPET

```
<div data-pv-component="content" data-pv-type="snippet" data-pv-id="343">
  {@page_content@}
  <span data-pv-element="page_content">This is where your content will display.</span>
</div>
```

```
<div id="owl-demo" class="owl-carousel">
  <div class="item" data-pv-component="content" data-pv-type="rotators" data-pv-parameters="{rotator_group:
'ghost'}">
    <div style="background-image: url({@rotator_image@}); height: 450px;">
      <div class="tier"
        <ul class="grid"="">
          <li class="grid-col grid-col_6of12 grid-col_push3of12 vr vr_x4">
            <h3 class="hdg hdg_3 mix-hdg_reversed" data-pv-element="rotator_title">
              Title
            </h3>
            <p class="txt txt_monospaced mix-txt_reversed" data-pv-element="rotator_brief">
              Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eligendi, repellendus repellat. Quo nemo
              quasi, ullam porro sequi voluptatum atque impedit inventore, iusto neque, ratione excepturi amet cumque veniam officia
              assumenda.
            </p>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>
```

BLOG/CONTENT

```
<div data-pv-component="content" data-pv-type="list" data-pv-parameters="{page_status: 'active', page_type: 'blog',
returntotal: 3, sortby: 'newest', from_date: '1/1/1900 12:00 AM', to_date: {@today@}}">
  <h2 data-pv-element="page_title">Page Title</h2>
  <span data-pv-element="page_content">This is where your content will display.</span>
  <a href="/page/{@marketing_keyword@}">View Post</a>
</div>
```

SCHEDULER OBJECT

```
<li data-pv-component="scheduler" data-pv-type="days" id="{@start_date:date(MMddyyyy)}" data-pv-
parameters="{slot_start:{@today@},slot_end:{@today:addDays(10)}" data-pv-object="schedule">
```

MEDIA

```
<div data-pv-component="media" data-pv-type="" data-pv-parameters="{sortby: 'date', media_type: 'image',
returnTotal: 10}">
  <h2 data-pv-element="media_title">Title of Image</h2>
  
  <p data-pv-element="media_description">Description of Image</p>
</div>
```

SURVEY

```
<div data-pv-component="survey" data-pv-type="info" data-pv-parameters="{survey_status:'active'}">
  <h3 data-pv-element="survey.survey_name">Survey Name</h3>
  <p data-pv-element="survey.survey_description">A brief description about the survey</p>
```



```

</div>
<div data-pv-component="survey" data-pv-type="questions" data-required="{@field_required@}">
  <label data-pv-element="field_label">Label</label>
  {@input@}
  <p class="help-block" data-pv-element="field_help"></p>
</div>

```

TEXT

```

<div data-pv-component="text">
  What sport do you like?
</div>
<div data-pv-component="text" data-pv-conditional="{@sport@} = baseball">
  Baseball
</div>
<div data-pv-component="text" data-pv-conditional="{@sport@} = basketball">
  Basketball
</div>

```

GIFT CARD

```

<div data-pv-component="giftcard" data-pv-type="" data-pv-parameters="{customer_no:{@customer_no@}}">
  Congratulations, you recieved a gift card in the amount of <span data-pv-formatter="currency" data-pv-
  element="current_value">$123.45</span> to shop online! To redeem your gift card visit <a
  href="http://yourcompany.com/giftcard">our website</a>.
</div>

```

```

<div data-pv-component="giftcard" data-pv-type="" data-pv-parameters="{customer_no: {@customer_no@},
  gift_card_no: {@reference_no@}}">
  <p>Dear <span data-pv-element="customer.first_name">First Name</span>,</p>
  <p>{@message@}</p>
  <p>
    <strong>From:</strong> <span data-pv-element="purchaser.first_name">First</span> <span data-pv-
    element="purchaser.last_name">Last</span>
    <br>
    <strong>Gift Card Value:</strong> <span data-pv-formatter="currency" data-pv-
    element="current_value">$100.00</span>
    <br>
    <strong>Your Gift Card Number is:</strong> <span data-pv-element="serial_number">123456789</span>
    <br>
    <strong>Your Pin # is:</strong> <span data-pv-element="gift_card_pin">1234</span>
  </p>
  <p>Your gift card is waiting in your online account. &nbsp;&nbsp;&nbsp;Click Go Shopping and your gift card will be waiting there for
  you to use.</p>
  <a href="{@domain_name@}/shop">Go Shopping</a>
</div>

```

SURVEY RESPONSE (data-pv-repeater)

```

<div align="left" style="text-align: left;" data-pv-component="survey" data-pv-type="response" data-pv-
  parameters="{customer_no: {@customer_no@}, survey_no: {@referenceType.survey@},
  survey_response_no: {@referenceType.survey_response@}}">
  <ul>
    <li data-pv-repeater="responses">
      <b style="font-weight:bold;" data-pv-element="field_label">Field Label</b>:
      <span data-pv-element="survey_answer">Response</span>
    </li>
  </ul>
</div>

```

<div>

MEMBERS

```
<li data-pv-component="membership" data-pv-type="members" data-pv-parameters="{sort_by: 'last_name'}">
  <span class="blockTile mix-blockTile_outlined">
    <a href="mailto:{@email@}" class="blockTile-hd blockTile-hd_flush">
      <span class="imgMask imgMask_fluid">
        
      </span>
    </a>
    <span class="blockTile-bd vr">
      <h6 class="hdg hdg_6 vr-override_x4">{@first_name@} {@last_name@}</h6>
      <h5 class="hdg hdg_5">{@company_name@}</h5>
    </span>
    <span class="blockTile-ft vr vr_x6">
      <dl class="defList defList_inline">
        <dt class="defList-term">Email</dt>
        <dd class="defList-desc">
          <a href="mailto:{@email@}"><span class="txt txt_feature" data-pv-element="email">Email Address</span>
        </a>
        </dd>
        <dt class="defList-term">Mobile</dt>
        <dd class="defList-desc">
          <a href="tel:{@mobile_number@}"><span class="txt txt_feature" data-pv-element="mobile_number" data-
pv-formatter="phone">PHONE</span></a>
        </dd>
      </dl>
    </span>
  </span>
</li>
```