

# Vidbaby API Solutions and Usage

## Introduction

This document will cover an overview of the solutions the avatar development library referred to in this document as Vidbaby API, can provide to customers as well as a basic understanding of how to implement them.

The target audience of this document are companies looking for a fully customizable solution that utilizes the Vidbaby API to create interactive avatars and the developers implementing it. It is recommended that the implementation of the api is done by one or more developers experienced with Javascript and asynchronous communication.

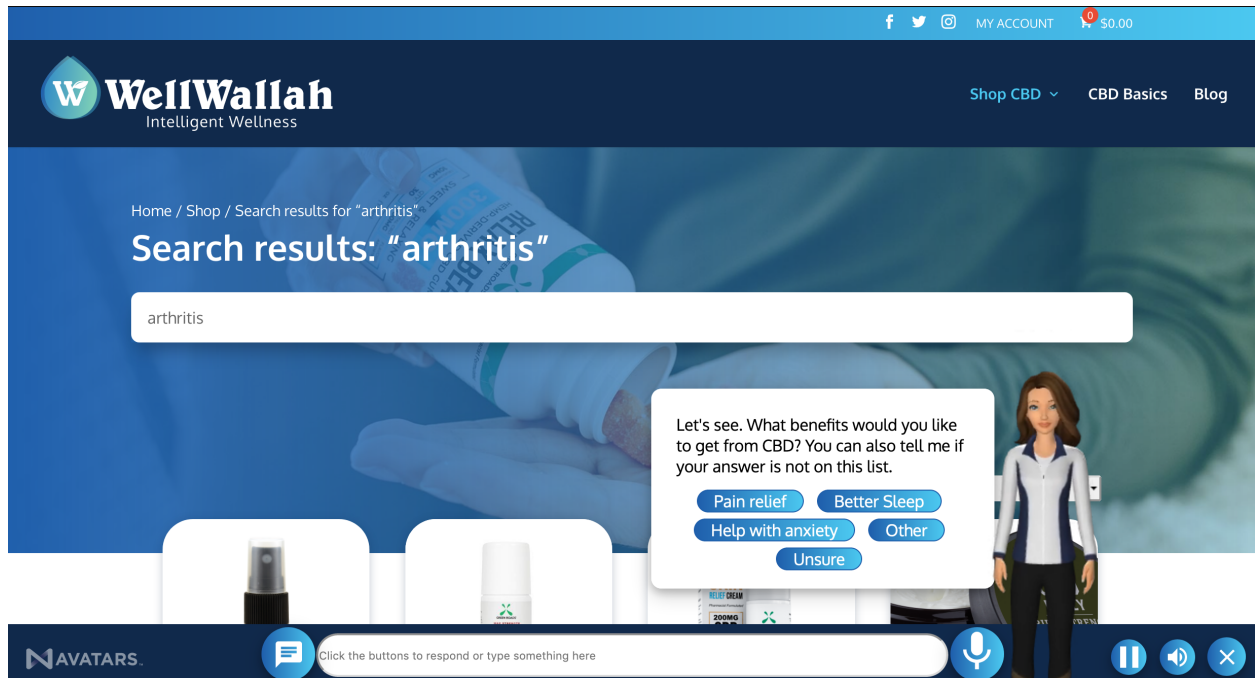
## Contents

Solutions	2
Webstore guide (wellwallah screenshot)	2
Classroom (un poc screenshot)	3
Interactivity	3
Natural Language Processing	3
Guide	4
Quick Start	4
Events	5
Examples	6
Initialize	6
Send Question	6
Get Response	6
Get Extracted Entity	6
Natural language processing and mic interaction	6

## Solutions

Our customers have found innovative ways to enhance their content with interactive avatars from the Vidbaby API. Below are a few examples of what can be done with this technology, but don't let that limit you. There are ways to use this technology that no one has thought of yet; send your ideas our way and our talented tech team will find new solutions to make it work.

## Webstore guide



The screenshot displays the WellWallah website interface. At the top, there is a navigation bar with social media icons (Facebook, Twitter, Instagram), a 'MY ACCOUNT' link, and a shopping cart icon showing a balance of \$50.00. The main header features the WellWallah logo with the tagline 'Intelligent Wellness' and navigation links for 'Shop CBD', 'CBD Basics', and 'Blog'. The main content area shows search results for 'arthritis', with a search bar containing the text 'arthritis'. A chat window is open, featuring a female avatar and a message: 'Let's see. What benefits would you like to get from CBD? You can also tell me if your answer is not on this list.' Below the message are five buttons: 'Pain relief', 'Better Sleep', 'Help with anxiety', 'Other', and 'Unsure'. At the bottom of the chat window, there is a text input field with the placeholder 'Click the buttons to respond or type something here' and a microphone icon. The 'NAVATARS.' logo is visible in the bottom left corner of the chat interface.

Create pre-made responses for questions and searches to help customers find exactly what products they are looking for. Avatar conversations can trigger events or navigation on your website enabling an interactive web store experience.

## Classroom



Create clickable responses that can help train staff and volunteers by guiding them through scenarios. Clickable responses are a series of predefined questions based on the previous response; clickable responses are often displayed in a list that the user can click on in order to send that text as a question to the avatar.

Track and score user responses by listening to response events.

### Interactivity

The possibilities are endless, whether you want an interactive appointment scheduling avatar for your doctor's office or an odds calculator for online betting platforms, the Vidbaby API can be used to create almost any predefined interaction you can imagine. Avatars understand context and can comprehend many different variations of what a user says, whereas "Do you have anything for arthritis?" and "What about products for arthritis?" will both be considered the same by the avatar.

### Natural Language Processing

We leverage Google's natural language processing in order to bring the next level of immersion to users. By triggering an event in the Vidbaby API you can tell the avatar to start listening for any words in any language and send that as a question to the avatar, then queuing a response based on the question.

## Guide

### Quick Start

Copying the code below into a HTML file will give you a working avatar and a basic starting point for your own custom UI:

```
<body>
  <div id="avatarContainerDiv"></div>
  <input /><button>Ask</button>
  <div class="messages"></div>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script type="text/javascript" src="https://avatar.codebaby.com/loader.js?id=12"></script>
  <script type="text/javascript">
    $('button').click(() => {
      let text = $('input').val();
      $('input').val("");
      $('.messages').append(`<p>${ text }</p>`);

      //send message to avatar
      $(document).trigger('na-ask.vidbaby', text);
    });

    //listen for response from avatar
    $(document).on('na-response.vidbaby', function (event, data) {
      $('.messages').append(`<p>${ data.answerText }</p>`);
    });
  </script>
</body>
```

## Events

Eventing makes up the backbone of the Vidbaby API, understanding these events and how to use them is the key to developing a production ready application or website with Vidbaby avatars.

- Listen for
  - **na-playerBuilt.vidbaby** - Vidbaby API, and models are fully loaded and ready to receive questions.
  - **na-playerLoadingAssets.vidbaby** - Vidbaby API is loaded and fully functional. However some assets (such as models) are still loading.
  - **na-extractedEntity.vidbaby** - Extracted entity like a search term. (for more about entities see the dialogflow documentation)
  - **na-clientData.vidbaby** - Triggered if there was a JS object called 'clientData' in the custom payload of the response from dialogflow.
  - **na-response.vidbaby** - Response from avatar after a text question, speech recognition, or predefined callback.
  - **na-segmentEnded.vidbaby** - The avatar has completed giving a response.
  - **na-microphoneData.vidbaby** - Data returned from natural language processor if mic is on.
  - **na-search.vidbaby** - Value of a search command response from the avatar.
  - **na-microphoneStarted.vidbaby** - Microphone is ready and listening
  - **na-microphoneStopped.vidbaby** - Microphone has stopped listening by either detecting the end of a phrase or na-microphoneOff was triggered.
- Trigger
  - **na-microphoneToggle.vidbaby** - Toggle microphone from on/off.
  - **na-microphoneOn.vidbaby** - Turn microphone on.
  - **na-microphoneOff.vidbaby** - Turn microphone off.
  - **na-pause.vidbaby** - Pause player
  - **na-play.vidbaby** - Resume playing after paused
  - **na-mute.vidbaby** - Mute avatar
  - **na-unmute.vidbaby** - Unmute avatar
  - **na-ask.vidbaby** - send question to avatar
  
  - **na-playResponse.vidbaby** - the avatar renders any string passed in data.answer
  - **na-playResponseData.vidbaby** - the avatar renders a response that contains a custom payload. The data passed to this event matches the data format 'Payload Content' section of the admin response creator with the additional element of data.answerText = "your avatar response text"
  
  -

## Examples

### Initialize

```
$(document).on('na-playerBuilt.vidbaby', function(event, data) {  
    //post initialisation code here.  
});
```

### Send Question

```
var question = "Hello, how are you?";  
$(document).trigger('na-ask.vidbaby', question);
```

### Get Response

```
$(document).on('na-response.vidbaby', function (event, responseData) {  
    //responseData contains all relevant information returned from the avatar. If you are  
    using the custom clientData object, that information is available in responseData.clientData  
    if(responseData.answerText) {  
        $('#text-display').append('<p>Avatar: '+responseData.answerText+'</p>');  
    }  
    //Clickable responses will be returned in responseData.clickableResponses  
    if(responseData.clickableResponses) {  
        addClickableResponses(responseData.clickableResponses);  
    }  
});
```

### Get Extracted Entity

//extracted entities like search terms, or names can be caught here. entity extraction can be setup in DialogFlow.

```
$(document).on('na-extractedEntity.vidbaby', function (event, data) {  
    console.log('na-extractedEntity', data);  
});
```

### Natural language processing and mic interaction

```
$('#button#microphone').click(function () {  
    //enable or disable the microphone from listening, here that is attached to a button  
    $(document).trigger('na-microphoneToggle.vidbaby');
```

```
});  
$(document).on('na-microphoneData.vidbaby', function (event, data) {  
    //audio is recognized as the user is speaking  
    console.log(data.text);  
    //After the user is done talking the message is have an isFinal flag set to true in the  
data  
    if(data.isFinal) {  
        //send message to avatar  
        $(document).trigger('na-ask', data.text);  
    }  
});
```

### **Make the Avatar Say Something**

```
var data = {};  
data.answer = 'Look, I can say something';  
vidbaby.$(document).trigger('na-playResponse.vidbaby', data);
```