

# Pets at Shelter API

The [Adopt-a-Pet.com](#) API will allow partners to query the [Adopt-a-Pet.com](#) database for information that can be used to build pet listings.

## API Features

Retrieve the list of available pets for the shelter or rescue

## API Design

The API is designed to receive inputs from your custom application using HTTP GET requests that are formatted with a script name and a querystring.

The results of the request will be a text result formatted in one of the available output styles. From there, you can use your own output parser to break the returned data into local data structures, and then process them as you want to.

Using the API functions provided, you should be able to build a pet list interface that is customized for your website's look and feel. You can also use the `pets_at_shelters` function (note that "shelters" is plural, unlike the `pets_at_shelter` function) to create a pet list for multiple shelters. This would be especially useful for organizations that have multiple branch locations.

## API Updates

October 1, 2025:

The `search_form` endpoint returns a new and improved list of breeds which matches the breed list on the Adopt a Pet website (and includes designer breeds such as Puggle, Aussiedoodle, Cockapoo, among others).

The limit for the maximum number of pets returned from the `pet_search` endpoint has been raised from 500 to 10,000. There is a limit of 500 pets that can be retrieved at one time using the pagination parameters `start_number` and `end_number`.

The `pet_search` endpoint now returns pets from Rehome, Adopt a Pet's private owner adoption program.

The `pet_search` endpoint now accepts a `cared_for_by` parameter, which filters pets by which types of organizations are caring for the given pets. Valid values are 'shelter', 'rescue', or 'private\_owner'.

The `pet_search` endpoint now accepts a `sort` parameter to adjust the ordering of the pet results. Valid values are 'nearest' or 'newest'.

The `pet_search` endpoint now returns higher quality, better optimized, and better cropped pet photos.

## Global Inputs

The following inputs can be submitted with every function call to the API. Globally required inputs are in **bold**.

API Key	key=	A string that identifies your partner account and prevents unauthorized use of the API.
API Version	v=	A string that indicates what version of the API you expect to be querying. The current version options are " <b>beta</b> ", " <b>1</b> " (the default) and " <b>2</b> ".

## Global Outputs

Each API query will include a response element called "**status**" whose value will be "**ok**" or "**fail**". If the status is "**fail**", we will also return error codes and error messages .

### Pet List Functions

Function Name	Inputs	Outputs	Notes
<code>pets_at_shelter</code>	<b>shelter_id</b>  start_number (default: 1)  end_number (default: 50, max: 25000)  meta_only (default: 0)	<ul style="list-style-type: none"><li>total_pets</li><li>returned_pets (this call)</li><li>pets<ul style="list-style-type: none"><li>order</li><li>results_photo_url</li><li>results_photo_width</li><li>results_photo_height</li><li>large_results_photo_url</li><li>large_results_photo_width</li><li>large_results_photo_height</li><li>details_url</li><li>species</li><li>pet_id</li><li>pet_name</li><li>sex</li><li>age</li><li>size</li><li>primary_breed</li></ul></li></ul>	If "meta_only" is passed as true, we'll only return result counts, not actual pet data. You can create your own pagination scheme by passing a start and ending number for the subset of results you want back.

		<ul style="list-style-type: none"> <li>• addr_city</li> <li>• addr_state_code</li> <li>• shelter_name</li> <li>• shelter_details_url</li> </ul>	
pets_at_shelter	<b>shelter_ids</b>  start_number (default: 1)  end_number (default: 50, max: 25000)  meta_only (default: 0)	<ul style="list-style-type: none"> <li>• total_pets</li> <li>• returned_pets (this call)</li> <li>• pets <ul style="list-style-type: none"> <li>• order</li> <li>• results_photo_url</li> <li>• results_photo_width</li> <li>• results_photo_height</li> <li>• large_results_photo_url</li> <li>• large_results_photo_width</li> <li>• large_results_photo_height</li> <li>• details_url</li> <li>• species</li> <li>• pet_id</li> <li>• pet_name</li> <li>• sex</li> <li>• age</li> <li>• size</li> <li>• primary_breed</li> <li>• addr_city</li> <li>• addr_state_code</li> <li>• shelter_name</li> <li>• shelter_details_url</li> </ul> </li> </ul>	If "meta_only" is passed as true, we'll only return result counts, not actual pet data. You can create your own pagination scheme by passing a start and ending number for the subset of results you want back.

## Sample Request - Response

Request pets\_at\_shelter:

/search/pets\_at\_shelter?key=bf650513c89d8132d9a1d2888973198d&shelter\_id=96812&start\_number=1&end\_number=70&output=json

Response:

```

{
  "status": "ok",
  "returned_pets": 9,
  "total_pets": 9,
  "pets": [
    {
      "order": 1,
      "results_photo_url": "https://pet-uploads-staging.adoptapet.com/null",
      "results_photo_width": null,
      "results_photo_height": null,
      "large_results_photo_url": "https://pet-uploads-staging.adoptapet.com/null",
      "large_results_photo_width": null,
      "large_results_photo_height": null,
      "details_url": "https://api-staging.adoptapet.com/search/pet_details?pet_id=28529460&key=bf650513c89d8132d9a1d2888973198d&v=1&output=json",
      "pet_id": "28529460",
      "pet_name": "Sasha",
      "sex": "f",
      "age": "adult",
      "size": "X-Large 101 lbs (46 kg) or more",
      "species": "dog",
      "primary_breed": "Rottweiler",
      "addr_city": "Atascadero",
      "addr_state_code": "CA",
      "secondary_breed": null
    },
    {
      "order": 2,
      "results_photo_url": "https://pet-uploads-staging.adoptapet.com/null",
      "results_photo_width": null,
      "results_photo_height": null,
      "large_results_photo_url": "https://pet-uploads-staging.adoptapet.com/null",
      "large_results_photo_width": null,
      "large_results_photo_height": null,
      "details_url": "https://api-staging.adoptapet.com/search/pet_details?pet_id=28529460&key=bf650513c89d8132d9a1d2888973198d&v=1&output=json",
      "pet_id": "28529460",
      "pet_name": "Sasha",
      "sex": "f",
      "age": "adult",
      "size": "X-Large 101 lbs (46 kg) or more",
      "species": "dog",
      "primary_breed": "Rottweiler",
      "addr_city": "Atascadero",
      "addr_state_code": "CA",
      "secondary_breed": null
    }
  ]
}

```

Request pets\_at\_shelters:

```
/search/pets_at_shelters?key=bf650513c89d8132d9a1d2888973198d&shelter_id=96814
&shelter_id=96812&start_number=1&end_number=70&output=json
```

Response:

```
{
  "status": "ok",
  "returned_pets": 12,
  "total_pets": 12,
  "pets": [
    {
      "order": 1,
      "results_photo_url": "https://pet-uploads-staging.adoptapet.com/null",
      "results_photo_width": null,
      "results_photo_height": null,
      "large_results_photo_url": "https://pet-uploads-staging.adoptapet.com/null",
      "large_results_photo_width": null,
      "large_results_photo_height": null,
      "details_url": "https://api-staging.adoptapet.com/search/pet_details?pet_id=28529460&key=bf650513c89d8132d9a1d2888973198d&v=1&output=json",
      "pet_id": "28529460",
      "pet_name": "Sasha",
      "sex": "f",
      "age": "adult",
      "size": "X-Large 101 lbs (46 kg) or more",
      "species": "dog",
      "primary_breed": "Rottweiler",
      "addr_city": "Atascadero",
      "addr_state_code": "CA",
      "secondary_breed": null
    },
    {
      "order": 2,
      "results_photo_url": "https://pet-uploads-staging.adoptapet.com/6/e/7/466182584.jpg",
      "results_photo_width": 70,
      "results_photo_height": 90,
      "large_results_photo_url": "https://pet-uploads-staging.adoptapet.com/a/5/d/466182587.jpg",
      "large_results_photo_width": 200,
      "large_results_photo_height": 200,
      "details_url": "https://api-staging.adoptapet.com/search/pet_details?pet_id=27750126&key=bf650513c89d8132d9a1d2888973198d&v=1&output=json",
      "pet_id": "27750126",
      "pet_name": "Diego",
      "sex": "m"
    }
  ]
}
```

## Error Codes

API function call responses will always always include a top level attribute of `status` and a value of `fail` in the event of an error. A nested `error` data structure will also be included in the response, and it has the following attributes:

- `code` – required, a numeric code from the table below
- `msg` – required, a technical name from the table below
- `details` – optional, additional free-form text explaining the error. For validation errors, we return a data structure with records for `"missing"`, `"invalid"`, and `"unknown"` elements.

The table below describes failure error codes that may be returned from API function calls:

Error Code	Error Message	Meaning
400	input_invalid	A field had invalid or badly formatted data
401	invalid_key	The API key submitted was invalid, or was submitted from an unauthorized domain

403	access_denied	Your partner account does not have permission to perform the request submitted
412	input_missing	A required field was not submitted in the request.
450	validation_failure	An error in the information submitted prevented the request from being completed. Check the <a href="#">missing, invalid, or unknown</a> attributes for further information.
500	tech_failure	The system was not able to complete your request for some other reason
503	resource_unavailable	The system has received too many requests from you and will not complete your request

## Example Error Code 450

Request:

```
/search/pets_at_shelters?key=bf650513c89d8132d9a1d2888973198d&start_number=1&end_number=30000000
```

Response:

```

{
  "status": "fail",
  "error": {
    "missing": [
      {
        "field": "shelter_id",
        "msg": "Please check your submission: it appears that a required field was not specified."
      }
    ],
    "code": 450,
    "details": "The method inputs didn't pass validation. Please check the missing, invalid, and unknown structures for more information.",
    "msg": "validation_failure"
  }
}

```