

---

# Inshapardaz API

*Release*

September 22, 2016



<b>1</b>	<b>API Key</b>	<b>3</b>
<b>2</b>	<b>Authorisation</b>	<b>5</b>
2.1	Authentication . . . . .	5
2.2	Getting Started . . . . .	5
2.3	Standard request and response headers . . . . .	5
2.4	Get Dictionaries . . . . .	5
2.5	Create Dictionary . . . . .	6
2.6	GET dictionary . . . . .	7
2.7	Update Dictionary . . . . .	7
2.8	Delete Dictionary . . . . .	8
2.9	Dictionary . . . . .	9
2.10	Entry . . . . .	9
2.11	Meaning . . . . .	9
2.12	Meaning Context . . . . .	9
2.13	Relationship . . . . .	9
2.14	Translation . . . . .	9
2.15	Word Details . . . . .	9



**Note**

This documentation is a work in progress.

Inshapardaz APIs are a way to consume inshapardaz data store. The basic aim this API to let any application access the processed entities stored in the application.

Service is a standard REST API using HATEAOS. The main entry point is the [entry] (entry) you can follow to get access to other resources.



---

**API Key**

---

Currently this api do not require any key but this can change in future.





---

## Authorisation

---

API will expose only the links that can be allowed for the user in context. If no user context is provided a guest user is assumed.

### 2.1 Authentication

### 2.2 Getting Started

### 2.3 Standard request and response headers

### 2.4 Get Dictionaries

Retrieves all dictionaries available to user.

#### 2.4.1 Request

**reqheader Accept** An **mime type** specifying the desired format of response. Supported formats are **application/json** and **application/xml**.

**reqheader Authorization** An OAuth2 token to authorise user. No token needed if the call is made as guest user.

#### 2.4.2 Response

Returns the list of all dictionaries available to user. Please review [Dictionary Resource](DictionaryResource) for details on the individual items in array.

#### 2.4.3 Response Codes

**code 200** Success



## 2.6 GET dictionary

### 2.6.1 Request

### 2.6.2 Request Parameters

Parameter Name	Data Type	Description
id   number		Id of dictionary to get

### 2.6.3 Response

### 2.6.4 Response Codes

**200** Success

**403** Unauthorised

**404** Dictionary not found

## 2.7 Update Dictionary

#### Uri

/api/dictionaries/{dictionaryId}

#### Request

```
` PUT /api/dictionaries/{id} HTTP/1.1 Content-Type: application/json
Authorization: OAuth2 ... `
```

“` javascript {

id : 12, name : 'dictionary name', description : 'dictionary description', language : 34, isPublic : true

### 2.7.1 }

##### Request Parameters

Parameter Name | Data Type | Description |

| id | number | Id of dictionary to update |

#### Response ` HTTP/1.1 200 OK Content-Type: application/json `

“` javascript {

id : 12, name : 'dictionary name', description : 'dictionary description', language : 34, isPublic : true

## 2.7.2 }

### ### Error Responses ###

Case	Response Code	Error Code
1	200	0
2	200	0
3	200	0
4	200	0
5	200	0
6	200	0
7	200	0
8	200	0
9	200	0
10	200	0
11	200	0
12	200	0
13	200	0
14	200	0
15	200	0
16	200	0
17	200	0
18	200	0
19	200	0
20	200	0
21	200	0
22	200	0
23	200	0
24	200	0
25	200	0
26	200	0
27	200	0
28	200	0
29	200	0
30	200	0
31	200	0
32	200	0
33	200	0
34	200	0
35	200	0
36	200	0
37	200	0
38	200	0
39	200	0
40	200	0
41	200	0
42	200	0
43	200	0
44	200	0
45	200	0
46	200	0
47	200	0
48	200	0
49	200	0
50	200	0
51	200	0
52	200	0
53	200	0
54	200	0
55	200	0
56	200	0
57	200	0
58	200	0
59	200	0
60	200	0
61	200	0
62	200	0
63	200	0
64	200	0
65	200	0
66	200	0
67	200	0
68	200	0
69	200	0
70	200	0
71	200	0
72	200	0
73	200	0
74	200	0
75	200	0
76	200	0
77	200	0
78	200	0
79	200	0
80	200	0
81	200	0
82	200	0
83	200	0
84	200	0
85	200	0
86	200	0
87	200	0
88	200	0
89	200	0
90	200	0
91	200	0
92	200	0
93	200	0
94	200	0
95	200	0
96	200	0
97	200	0
98	200	0
99	200	0
100	200	0

dictioanry | 403 Forbidden | Unauthorised | | No matching dictionary found | 404 Not Found | Dictionary Not Found | | Title missing | 400 Bad Request | Title required | | Language id not valid | 400 Bad Request | Invlaid language | | Posted data not correct format | 400 Bad Request | Invlaid data |

### ### Delete dictionary

```
#### Uri /api/dictionaries/{id}
```

#### #### Request

```
` DELETE /api/dictionaries/{id} HTTP/1.1 Content-Type: application/json
Authorization: OAuth2 ... `
```

## ##### Request Parameters

Parameter Name	Data Type	Description
name	String	Full name of the person
age	Integer	Age in years
gender	String	Gender (Male/Female)
email	String	Email address
phone	String	Phone number
password	String	Secure password
username	String	Unique username
role	String	User role (Admin/User)
status	String	Account status (Active/Inactive)
last_login	Timestamp	Last login time
created_at	Timestamp	Account creation time
updated_at	Timestamp	Last update time

|-----|-----|-----| id | number | Id of dictionary to delete

```
#### Response ` HTTP/1.1 204 NoContent Content-Type: application/json `
```

### ### Error Responses ###

Case	Response Code	Error Code
1	200	
2	200	
3	200	
4	200	
5	200	
6	200	
7	200	
8	200	
9	200	
10	200	
11	200	
12	200	
13	200	
14	200	
15	200	
16	200	
17	200	
18	200	
19	200	
20	200	
21	200	
22	200	
23	200	
24	200	
25	200	
26	200	
27	200	
28	200	
29	200	
30	200	
31	200	
32	200	
33	200	
34	200	
35	200	
36	200	
37	200	
38	200	
39	200	
40	200	
41	200	
42	200	
43	200	
44	200	
45	200	
46	200	
47	200	
48	200	
49	200	
50	200	
51	200	
52	200	
53	200	
54	200	
55	200	
56	200	
57	200	
58	200	
59	200	
60	200	
61	200	
62	200	
63	200	
64	200	
65	200	
66	200	
67	200	
68	200	
69	200	
70	200	
71	200	
72	200	
73	200	
74	200	
75	200	
76	200	
77	200	
78	200	
79	200	
80	200	
81	200	
82	200	
83	200	
84	200	
85	200	
86	200	
87	200	
88	200	
89	200	
90	200	
91	200	
92	200	
93	200	
94	200	
95	200	
96	200	
97	200	
98	200	
99	200	
100	200	

dictioanry | 403 Forbidden | Unauthorised | | No matching dictionary found | 404 Not Found | Dictionary Not Found |

## 2.8 Delete Dictionary

```
#### Uri /api/dictionaries/{id}
```

#### #### Request

```
` DELETE /api/dictionaries/{id} HTTP/1.1 Content-Type: application/json
Authorization: OAuth2 ... `
```

## ##### Request Parameters

Parameter Name	Data Type	Description
name	String	Full name of the person
age	Integer	Age of the person
gender	String	Gender of the person
email	String	Email address of the person
phone	String	Phone number of the person
address	String	Address of the person
city	String	City of the person
state	String	State of the person
zip	String	Zip code of the person
password	String	Password of the person
username	String	Username of the person
role	String	Role of the person
status	String	Status of the person
created_at	Timestamp	Timestamp when the record was created
updated_at	Timestamp	Timestamp when the record was last updated

---

| id | number | Id of dictionary to delete |

#### Response ` HTTP/1.1 204 NoContent Content-Type: application/json `

### Error Responses ###

**Case | Response Code | Error Code |**

---

| User not allowed to delete dictionary | *403 Forbidden* | Unauthorised | | No matching dictionary found | *404 Not Found* | Dictionary Not Found |

## 2.9 Dictionary

### 2.10 Entry

### 2.11 Meaning

### 2.12 Meaning Context

### 2.13 Relationship

### 2.14 Translation

### 2.15 Word Details



## C

CreateDictionary, 5

## D

DeleteDictionary, 8

## G

GetDictionaries, 5

GetDictionary, 6

## R

requests

Dictionary, 5–8

## U

UpdateDictionary, 7