



RobotEvents API

Documentation

Access Tokens

Public Robot Events API

1.0.21 OAS 3.0

/api/v2/swagger.yml

An API to access data on Robot Events officially.

Request Metadata

Pagination is performed the same way throughout each pageable endpoint using query parameters **page** and **per_page** (per_page default is 25 with a maximum of 250).

Please note that dates should be formatted according to RFC3339.

Servers

https://www.robotevents.com/api/v2

Authorize



Event



GET /events



Gets a List of Events

Parameters

Try it out

Name	Description
id[]	Filter by Event ID
array[integer]	
(query)	
sku[]	Filter by Event SKU
array[string]	
(query)	
team[]	Filter by Teams that participated in the Event
array[integer]	
(query)	
season[]	Filter by the Season that the Event belonged to
array[integer]	
(query)	
start	Filter by the Start Date of the Event
string(\$date-time)	
(query)	<input type="text" value="start"/>
end	Filter by the End Date of the Event
string(\$date-time)	

Name	Description
(query)	<input type="text" value="end"/>
region	Filter by the region of the event
string	<input type="text" value="region"/>
(query)	
level[]	Filter by the Event Level
array[string]	Available values : World, National, State, Signature, Other
(query)	<div>-- World National State Signature Other</div>
myEvents	Only show events that have at least one registered team associated with the authenticated user.
boolean	Default value : false
(query)	<input type="text" value="false"/>
eventTypes[]	Filter by the Event Type
array[string]	Available values : tournament, league, workshop, virtual
(query)	<div>-- tournament league workshop virtual</div>

Responses

Code	Description	Links
200	List of Events	No links
Media type		
<div>application/json</div>		
Controls Accept header.		
Example Value Schema		
<pre>{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0, "name": "string" }] }</pre>		

GET /events/{id}

Gets a Single Event

Parameters

Try it out

Name	Description
id <small>* required</small>	The ID of the Event
integer(\$int32) (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	A single Event	No links
<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 0, "sku": "string", "name": "string", "start": "2025-04-05T2 0:54:38.634Z", "end": "2025-04-05T20:5 4:38.634Z", "season": { "id": 0, "name": "string", "code": "string" }, "program": { "id": 0, "name": "string", "code": "string" }, "location": { "venue": "string", "address_1": "strin "</pre></div>		
404	Specified Event not found	No links
<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "code": 0, "message": "string" }</pre></div>		

GET /events/{id}/teams

Gets a List of Teams present at a given Event

Parameters

Try it out

Name	Description
------	-------------

Name	Description
id <small>★ required</small>	The ID of the Event
integer(\$int32) (path)	<input type="text" value="id"/>
number[] array[string] (query)	Filter by Team Number
registered boolean (query)	Filter by whether the Team is Registered <input type="text" value="--"/>
grade[] array[string] (query)	Filter by the Grade of the Team Available values : College, High School, Middle School, Elementary School <div><input type="text" value="--"/> College High School Middle School Elementary School</div>
country[] array[string] (query)	Filter by the Country of the Team
myTeams boolean (query)	Only show teams associated with the authenticated user. Default value : false <input type="text" value="false"/>

Responses

Code	Description	Links
200	List of Teams	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "meta": {
    "current_page": 0,
    "first_page_url": "string",
    "from": 0,
    "last_page": 0,
    "last_page_url": "string",
    "next_page_url": "string",
    "path": "string",
    "per_page": 0,
    "prev_page_url": "string",
    "to": 0,
    "total": 0
  },
  "data": [
    {
      "id": 0,
      "number": "string"
    }
  ]
}
```

GET /events/{id}/skills

Gets a List of Skills runs performed at a given Event

Parameters

Try it out

Name	Description
id * required integer(\$int32) (path)	The ID of the Event <input type="text" value="id"/>
team[] array[integer] (query)	Filter by Team Number that performed the Skills run
type[] array[string] (query)	Filter by Type of Skills run Available values : driver, programming <div>-- driver programming</div>

Responses

Code	Description	Links
200	<div>A list of Skills</div> <div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <pre>{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0, "event": "s</pre>	

 No links |

GET /events/{id}/awards

Gets a List of Awards at a given Event

Parameters

Try it out

Name	Description
id <small>★ required</small>	The ID of the Event
integer(\$int32) (path)	<input type="text" value="id"/>
team[]	Filter by Team Number that the Award was awarded to
array[integer] (query)	
winner[]	Filter by the Winner of the Award (can include people's names, etc.)
array[string] (query)	

Responses

Code	Description	Links
200	A list of Awards	No links

Media type

application/json

Controls **Accept** header.

Example Value

Schema

```
{
  "meta": {
    "current_page": 0,
    "first_page_url": "string",
    "from": 0,
    "last_page": 0,
    "last_page_url": "string",
    "next_page_url": "string",
    "path": "string",
    "per_page": 0,
    "prev_page_url": "string",
    "to": 0,
    "total": 0
  },
  "data": [
    {
      "id": 0,
      "event": 0
    }
  ]
}
```

GET

/events/{id}/divisions/{div}/matches

Gets a List of Matches for a single Division of an Event

Parameters

Try it out

Name	Description
id <small>★ required</small>	The ID of the Event
integer(\$int32) (path)	<input type="text" value="id"/>

Name	Description
div <small>★ required</small>	The ID of the Division
integer(\$int32) (path)	<div>div</div>
team[] array[integer] (query)	Filter to only return Matches which involved given Teams
round[] array[integer] (query)	Filter by the Round of the Match. Some typical values are shown below: <ul style="list-style-type: none">• 1 - Practice• 2 - Qualification• 3 - Quarter-Finals• 4 - Semi-Finals• 5 - Finals• 6 - Round of 16• etc.
instance[] array[integer] (query)	Filter by the Instance of the Match. This is used to describe which Quarter-Final match (for example) is being played
matchnum[] array[integer] (query)	Filter by the MatchNum of the Match. This is the actual Match "number", e.g. Qualification Match, or the individual match in a Best of 3

Responses

Code	Description	Links
200	A list of Matches	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "meta": {
    "current_page": 0,
    "first_page_url": "string",
    "from": 0,
    "last_page": 0,
    "last_page_url": "string",
    "next_page_url": "string",
    "path": "string",
    "per_page": 0,
    "prev_page_url": "string",
    "to": 0,
    "total": 0
  },
  "data": [
    {
      "id": 0,
      "event": 5
    }
  ]
}
```

GET

/events/{id}/divisions/{div}/finalistRankings

Gets a List of Finalist Rankings for a single Division of an Event

Parameters

Try it out

Name	Description
id <small>★ required</small>	The ID of the Event
<code>integer(\$int32)</code> <i>(path)</i>	<input type="text" value="id"/>
div <small>★ required</small>	The ID of the Division
<code>integer(\$int32)</code> <i>(path)</i>	<input type="text" value="div"/>
<code>team[]</code> <code>array[integer]</code> <i>(query)</i>	Filter to only return Rankings which involve given Teams
<code>rank[]</code> <code>array[integer]</code> <i>(query)</i>	Filter by the Rank

Responses

Code	Description	Links
200	<div>A list of Rankings</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0, "event": 0 }]}</pre></div>	

 No links |

GET

/events/{id}/divisions/{div}/rankings

Gets a List of Rankings for a single Division of an Event

Parameters

Try it out

Name	Description
id * required integer(\$int32) (path)	The ID of the Event <input type="text" value="id"/>
div * required integer(\$int32) (path)	The ID of the Division <input type="text" value="div"/>
team[] array[integer] (query)	Filter to only return Rankings which involve given Teams
rank[] array[integer] (query)	Filter by the Rank

Responses

Code	Description	Links
200	A list of Rankings <div>Media type <div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><pre>{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0, "event": 0 }]}</pre></div>	No links

Team

GET /teams

🔒 ^

Gets a List of Teams

Parameters

Try it out

Name	Description
------	-------------

Name	Description
id[] array[integer] (query)	Filter by Team ID
number[] array[string] (query)	Filter by Team Number
event[] array[integer] (query)	Filter by Events that Teams have attended
registered boolean (query)	Filter by whether or not the Team is Registered <div>--</div>
program[] array[integer] (query)	Filter by the Program that the Team is part of
grade[] array[string] (query)	Filter by the Grade of the Team Available values : College, High School, Middle School, Elementary School <div>-- College High School Middle School</div>
country[] array[string] (query)	Filter by the Country of the Team
myTeams boolean (query)	Only show teams associated with the authenticated user. Default value : false <div>false</div>

Responses

Code	Description	Links
200	List of Teams <div>Media type application/json Controls Accept header.</div> <div>Example Value Schema</div> <div>{ "meta": { "current_page": 0, "first_page_url": "st ring", "from": 0, "last_page": 0, "last_page_url": "str ing", "next_page_url": "str ing", "path": "string", "per_page": 0, "prev_page_url": "str ing", "to": 0, }, }</div>	No links

CodeDescriptionLinks

```
"total": 0
},
"data": [
  {
    "id": 0
```

GET /teams/{id}

Gets a Single Team

ParametersTry it out

NameDescription

id * requiredThe ID of the Team

integer(\$int32)(path)id

Responses

CodeDescriptionLinks

200A single TeamNo links

Media typeapplication/jsonControls Accept header.Example ValueSchema

```
{
  "id": 0,
  "number": "string",
  "team_name": "string",
  "robot_name": "string",
  "organization": "string",
  "location": {
    "venue": "string",
    "address_1": "string",
    "address_2": "string",
    "city": "string",
    "region": "string",
    "postcode": "string",
    "country": "string",
    "coordinates": {
      "lat": 0,
      "lon": 0
    }
  }
}
```

404Specified Team not foundNo links

Media typeapplication/jsonExample ValueSchema



```
{
  "code": 0,
  "message": "string"
}
```

11 of 26

4/5/25, 1:54 PM

GET

/teams/{id}/events



Gets a List of Events that a given Team has attended

Parameters

Try it out

Name	Description
id <small>★ required</small>	The ID of the Team
integer(\$int32) (path)	<input type="text" value="id"/>
sku[]	Filter by Event SKU
array[string] (query)	
season[]	Filter by the Season that the Event belonged to
array[integer] (query)	
start	Filter by the Start Date of the Event
string(\$date-time) (query)	<input type="text" value="start"/>
end	Filter by the End Date of the Event
string(\$date-time) (query)	<input type="text" value="end"/>
level[]	Filter by the Event Level
array[string] (query)	Available values : World, National, State, Signature, Other
	<div>-- World National State Signature Other</div>

Responses

Code	Description	Links
200	List of Events	No links

Media type

application/json

Controls Accept header.

Example Value

Schema

```
{
  "meta": {
    "current_page": 0,
    "first_page_url": "st
ring",
    "from": 0,
    "last_page": 0,
    "last_page_url": "str
ing",
    "next_page_url": "str
ing",
    "path": "string",
    "per_page": 0,
  }
}
```

Code

Description

Links

```
"prev_page_url": "string",
"to": 0,
"total": 0
},
"data": [
{
"id": 0,
```

GET

/teams/{id}/matches

 ^

Gets a List of Matches that a given Team has played in

Parameters

Try it out

Name	Description
id <small>* required</small> integer(\$int32) (path)	The ID of the Team <div>id</div>
event[] array[integer] (query)	Filter by the Event the Match was performed at
season[] array[integer] (query)	Filter by the Season during which the Match was played
round[] array[integer] (query)	Filter by the Round of the Match. Some typical values are shown below: <ul style="list-style-type: none">1 - Practice2 - Qualification3 - Quarter-Finals4 - Semi-Finals5 - Finals6 - Round of 16etc.
instance[] array[integer] (query)	Filter by the Instance of the Match. This is used to describe which Quarter-Final match (for example) is being played
matchnum[] array[integer] (query)	Filter by the MatchNum of the Match. This is the actual Match "number", e.g. Qualification Match, or the individual match in a Best of 3

Responses

Code	Description	Links
200	List of Matches <div>Media type application/json <small>Controls Accept header.</small></div>	No links

Code	Description	Links
<div>Example Value Schema</div> <pre>{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0, "event": f</pre>		

GET /teams/{id}/rankings

Gets a List of Rankings for a given Team

Parameters

Try it out

Name	Description
id <small>* required</small> <code>integer(\$int32)</code> <i>(path)</i>	The ID of the Team <div>id</div>
<code>event[]</code> <code>array[integer]</code> <i>(query)</i>	Filter by the Event in which the Ranking was achieved
<code>rank[]</code> <code>array[integer]</code> <i>(query)</i>	Filter by the Rank
<code>season[]</code> <code>array[integer]</code> <i>(query)</i>	Filter by the Season during which the Ranking was

Responses

Code	Description	Links
200	List of Rankings <div>Media type<div>application/json</div><div>Controls Accept header.</div></div> <div>Example Value Schema</div> <pre>{ "meta": { "current_page": 0,</pre>	No links

Code	Description	Links
	<pre>"first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0,</pre>	

GET

/teams/{id}/skills

Gets a List of Skills runs that a given Team has performed

Parameters

Try it out

Name	Description
id <small>* required</small>	The ID of the Team
<small>integer(\$int32)</small> <small>(path)</small>	<input type="text" value="id"/>
<small>event[]</small> <small>array[integer]</small> <small>(query)</small>	Filter by the Event at which the Skills run was performed
<small>type[]</small> <small>array[string]</small> <small>(query)</small>	Filter by Type of Skills run <i>Available values</i> : driver, programming
	<div>-- driver programming</div>
<small>season[]</small> <small>array[integer]</small> <small>(query)</small>	Filter by the Season during which the Skills run was performed

Responses

Code	Description	Links
200	A list of Skills	No links

Media type

application/json

Controls Accept header.

Example Value

Schema

```
{
  "meta": {
    "current_page": 0,
```

Code	Description	Links
	<pre>"first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0</pre>	

GET

/teams/{id}/awards

Gets a List of Awards that a given Team has received

Parameters

Try it out

Name	Description
id <small>* required</small>	The ID of the Team
integer(\$int32) (path)	<input type="text" value="id"/>
event[] array[integer] (query)	Filter by the Event at which the Award was given out
season[] array[integer] (query)	Filter by the Season in which the Award was given out

Responses


Code	Description	Links
200	List of Awards	No links
	<div>Media type<div>application/json</div><div>Controls Accept header.</div><div>Example ValueSchema<pre>{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0,</pre></div></div>	

Code	Description	Links
	<pre>"prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0.</pre>	

Program ^

GET

/programs/{id}

 ^

Find a single Program by ID

Parameters

Try it out



Name	Description
id ★ required	The Program ID
integer(\$int32) (path)	<div>id</div>

Responses

Code	Description	Links
200	Single Program	No links
<div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 0, "abbr": "string", "name": "string" }</pre></div>		

GET

/programs



Gets a List of Programs

Parameters

Try it out

Name	Description
id[]	Filter by program ID
array[integer] (query)	

Responses

Code	Description	Links
200	List of Programs	No links

Media type

application/json

Controls **Accept** header.

Example Value Schema

```
{
  "meta": {
    "current_page": 0,
    "first_page_url": "string",
    "from": 0,
    "last_page": 0,
    "last_page_url": "string",
    "next_page_url": "string",
    "path": "string",
    "per_page": 0,
    "prev_page_url": "string",
    "to": 0,
    "total": 0
  },
  "data": [
    {
      "id": 0,
      "name": "string"
    }
  ]
}
```

Season

^

GET /seasons  ^

Gets a List of Seasons

Parameters

Try it out

Name	Description
id[]	Filter by Season ID
array[integer]	
(query)	
program[]	Filter by Program to which the
array[integer]	Season belongs
(query)	
team[]	Filter by seasons in which the
array[integer]	specified teams were active
(query)	
start	Filter by Start Date of the Season
string(\$date-time)	<div>start</div>
(query)	
end	Filter by End Date of the Season
string(\$date-time)	<div>end</div>
(query)	

Name

Description

active

Only include active Seasons

boolean

--

(query)

Responses

Code

Description

Links

200

List of Seasons

No links

Media type

application/json

Controls Accept header.

Example Value


Schema


```
{
  "meta": {
    "current_page": 0,
    "first_page_url": "string",
    "from": 0,
    "last_page": 0,
    "last_page_url": "string",
    "next_page_url": "string",
    "path": "string",
    "per_page": 0,
    "prev_page_url": "string",
    "to": 0,
    "total": 0
  },
  "data": [
    {
      "id": 0,
      "name": "season"
    }
  ]
}
```

GET /seasons/{id}

GET

/seasons/{id}





Gets a single Season

Parameters

Try it out

Name

Description

id * required

The Season ID

integer(\$int32)

id

(path)

Responses

Code

Description

Links

200

A single Season

No links

Media type

application/json

Controls Accept header.

Code	Description	Links
------	-------------	-------



Example Value

Schema

```
{
  "id": 0,
  "name": "string",
  "program": {
    "id": 0,
    "name": "string",
    "code": "string"
  },
  "start": "2025-04-05T20:54:38.687Z",
  "end": "2025-04-05T20:54:38.687Z",
  "years_start": 0,
  "years_end": 0
}
```

GET

/seasons/{id}/events



Gets a List of Events for a given Season

Parameters

Try it out

Name	Description
id <small>★ required</small>	The Season ID
integer(\$int32) (path)	<div>id</div>
sku[] array[string] (query)	Filter by Event SKU
team[] array[integer] (query)	Filter by Teams that participated in the Event
start string(\$date-time) (query)	Filter by the Start Date of the Event <div>start</div>
end string(\$date-time) (query)	Filter by the End Date of the Event <div>end</div>
level[] array[string] (query)	Filter by the Event Level Available values : World, National, State, Other <div>-- World National State Other</div>

Responses

Code	Description	Links
------	-------------	-------

20 of 26

4/5/25, 1:54 PM

Code	Description	Links
200	List of Events	No links
Media type		
application/json		
Controls Accept header.		
Example Value Schema		
{ "meta": { "current_page": 0, "first_page_url": "string", "from": 0, "last_page": 0, "last_page_url": "string", "next_page_url": "string", "path": "string", "per_page": 0, "prev_page_url": "string", "to": 0, "total": 0 }, "data": [{ "id": 0, "sku": "string" }] }		

Schemas	
Event { id* [...] sku* [...] name* [...] start [...] end [...] season* IdInfo {...} program* IdInfo {...} location* Location {...} locations [...] divisions [...] level EventLevel ongoing [...] awards_finalized [...] event_type EventType [...] }	
EventType string Enum: Array [4]	

```
Program {
  id          [...]
  abbr        [...]
  name        [...]
}

EventLevel string
Enum:
  Array [ 6 ]

Location {
  venue          [...]
  address_1      [...]
  address_2      [...]
  city           [...]
  region         [...]
  postcode       [...]
  country        [...]
  coordinates    Coordinates
                  {...}
}

Coordinates {
  lat          [...]
  lon          [...]
}

Locations {
  < * >:      Location
               {...}
}

Division {
  id          [...]
  name        [...]
  order       [...]
}

Grade string
Enum:
  Array [ 4 ]
```

```

Team {
  id* [...]
  number* [...]
  team_name [...]
  robot_name [...]
  organization [...]
  location Location {...}
  registered [...]
  program* IdInfo {...}
  grade Grade [...]
}

```

```

IdInfo {
  id* [...]
  name* [...]
  code [...]
}

```

```

MatchObj {
  id* [...]
  event* IdInfo {...}
  division* IdInfo {...}
  round* [...]
  instance* [...]
  matchnum* [...]
  scheduled [...]
  started [...]
  field [...]
  scored* [...]
  name* [...]
  alliances* [...]
}

```

```

Alliance {
  color* [...]
  score* [...]
  teams* [...]
}

```

```

AllianceTeam {
  team IdInfo {...}
  sitting [...]
}

```

```
Ranking {
  id [...]
  event IdInfo
    {...}
  division IdInfo
    {...}
  rank [...]
  team IdInfo
    {...}
  wins [...]
  losses [...]
  ties [...]
  wp [...]
  ap [...]
  sp [...]
  high_score [...]
  average_points [...]
  total_points [...]
}
```

```
Skill {
  id [...]
  event IdInfo
    {...}
  team IdInfo
    {...}
  type SkillType
    [...]
  season IdInfo
    {...}
  division IdInfo
    {...}
  rank [...]
  score [...]
  attempts [...]
}
```

```
SkillType string
Enum:
  Array [ 3 ]
```

```
Award {
  id [...]
  event IdInfo
    {...}
  order [...]
  title [...]
  qualifications [...]
  designation [...]
  classification [...]
  teamWinners [...]
  individualWinners [...]
}
```



```
TeamAwardWinner {
  division      IdInfo
                {...}
  team          IdInfo
                {...}
}

Season {
  id            [...]
  name          [...]
  program       IdInfo
                {...}
  start         [...]
  end           [...]
  years_start   [...]
  years_end     [...]
}

Error {
  code          [...]
  message       [...]
}

PageMeta {
  current_page  [...]
  first_page_url [...]
  from          [...]
  last_page     [...]
  last_page_url [...]
  next_page_url [...]
  path          [...]
  per_page      [...]
  prev_page_url [...]
  to            [...]
  total         [...]
}

PaginatedTeam {
  meta          PageMeta
                {...}
  data          [...]
}

PaginatedEvent {
  meta          PageMeta
                {...}
  data          [...]
}
```

PaginatedAward	{
meta	PageMeta
	{...}
data	[...]
	}
PaginatedSeason	{
meta	PageMeta
	{...}
data	[...]
	}
PaginatedRanking	{
meta	PageMeta
	{...}
data	[...]
	}
PaginatedMatch	{
meta	PageMeta
	{...}
data	[...]
	}
PaginatedSkill	{
meta	PageMeta
	{...}
data	[...]
	}
PaginatedProgram	{
meta	PageMeta
	{...}
data	[...]
	}

Robotics Education & Competition Foundation
1519 INTERSTATE HIGHWAY 30 W
Greenville, TX 75402
Email: support@robotevents.com

© 2025 Robotics Education & Competition Foundation.
All rights reserved. All other names/ trademarks are the property of their respective owners.
[Privacy Policy](#) | [Terms of Service](#) | [Acceptable Use Policy](#) | [Refund Policy](#)