

API Module - LIR

LIR Management



ProVision's APIv1 system has been replaced by APIv2, and is now considered deprecated.

- LIR Management
 - Get
 - Delete
 - Add
 - Update

Get

URL	/api/v1/api.php?target=lir&action=get
Description	Returns a list of LIRs

Returns	Examples: <table> <tr> <td>SUCCESSFUL</td><td> <pre>{ "success": 1, "message": "2 objects found", "data": [{ "id": "100", "name": "RIPE Test LIR", "slug": "ripe-test-lir", "entities": [{ "mnt_by": "mntner@email.com", "mnt_by_password": "password", "admin_c": "test-admin-c", "tech_c": "test-tech-c", "api_key": null }], "rir": "RIPE" }, { "id": "101", "name": "ARIN Test LIR", "slug": "arin-test-lir", "entities": [{ "org_handle": "TEST-10", "admin_poc": "TEST-ARIN", "net_poc": "TEST-ARIN", "abuse_poc": "", "net_name_prefix": "PRFX", "api_key": "API-XXXX-YYYY-ZZZZ-1234" }], "rir": "ARIN", "asn": "1000" }] }</pre> </td></tr> <tr> <td>ERROR</td><td> <pre>{ "success": 0, "message": "error message" }</pre> </td></tr> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "2 objects found", "data": [{ "id": "100", "name": "RIPE Test LIR", "slug": "ripe-test-lir", "entities": [{ "mnt_by": "mntner@email.com", "mnt_by_password": "password", "admin_c": "test-admin-c", "tech_c": "test-tech-c", "api_key": null }], "rir": "RIPE" }, { "id": "101", "name": "ARIN Test LIR", "slug": "arin-test-lir", "entities": [{ "org_handle": "TEST-10", "admin_poc": "TEST-ARIN", "net_poc": "TEST-ARIN", "abuse_poc": "", "net_name_prefix": "PRFX", "api_key": "API-XXXX-YYYY-ZZZZ-1234" }], "rir": "ARIN", "asn": "1000" }] }</pre>	ERROR	<pre>{ "success": 0, "message": "error message" }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "2 objects found", "data": [{ "id": "100", "name": "RIPE Test LIR", "slug": "ripe-test-lir", "entities": [{ "mnt_by": "mntner@email.com", "mnt_by_password": "password", "admin_c": "test-admin-c", "tech_c": "test-tech-c", "api_key": null }], "rir": "RIPE" }, { "id": "101", "name": "ARIN Test LIR", "slug": "arin-test-lir", "entities": [{ "org_handle": "TEST-10", "admin_poc": "TEST-ARIN", "net_poc": "TEST-ARIN", "abuse_poc": "", "net_name_prefix": "PRFX", "api_key": "API-XXXX-YYYY-ZZZZ-1234" }], "rir": "ARIN", "asn": "1000" }] }</pre>				
ERROR	<pre>{ "success": 0, "message": "error message" }</pre>				
Example URL	/api/v1/api.php?target=lir&action=get				

Delete

URL	/api/v1/api.php?target=lir&action=delete&id=ID
Description	Deletes an LIR

Returns	Examples: <table> <tr> <td>SUCCESSFUL</td><td> <pre>{ "success": 1, "message": "LIR deleted." }</pre> </td></tr> <tr> <td>ERROR</td><td> <pre>{ "success":0, "message":"error message" }</pre> </td></tr> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "LIR deleted." }</pre>	ERROR	<pre>{ "success":0, "message":"error message" }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "LIR deleted." }</pre>				
ERROR	<pre>{ "success":0, "message":"error message" }</pre>				
Example URL	/api/v1/api.php?target=lir&action=delete&id=100				

Add

URL	/api/v1/api.php?target=lir&action=add		
Description	Adds a new LIR. LIR creation utilizes the Resource system - see API Module - Resource - Add for details on adding a resource via the API.		
Returns	Examples: <table> <tr> <td>SUCCESSFUL</td><td> <pre>{ "success":1, "message":"Resource added", "data": { "id":14063, "name":"TestLIR", "slug":"testlir", "type":"entry", "parent_id":1, "category_id":null, "date":1517513319, "modified":1517513319, "attr": { "_section":"94", "rir":"ARIN", "asn":"20202", "org-handle":"org1", "admin-poc":"admin1", "net-poc":"tech1", "abuse-poc":"abuse1", "net-name-prefix":"test-", "api-key":10, "mnt-by":"", </pre> </td></tr> </table>	SUCCESSFUL	<pre>{ "success":1, "message":"Resource added", "data": { "id":14063, "name":"TestLIR", "slug":"testlir", "type":"entry", "parent_id":1, "category_id":null, "date":1517513319, "modified":1517513319, "attr": { "_section":"94", "rir":"ARIN", "asn":"20202", "org-handle":"org1", "admin-poc":"admin1", "net-poc":"tech1", "abuse-poc":"abuse1", "net-name-prefix":"test-", "api-key":10, "mnt-by":"", </pre>
SUCCESSFUL	<pre>{ "success":1, "message":"Resource added", "data": { "id":14063, "name":"TestLIR", "slug":"testlir", "type":"entry", "parent_id":1, "category_id":null, "date":1517513319, "modified":1517513319, "attr": { "_section":"94", "rir":"ARIN", "asn":"20202", "org-handle":"org1", "admin-poc":"admin1", "net-poc":"tech1", "abuse-poc":"abuse1", "net-name-prefix":"test-", "api-key":10, "mnt-by":"", </pre>		

	<pre>"admin-c":""," "tech-c":""," "_gadget_data":false }, "section": { "id":"94", "name":"LIR", "slug":"lir", "type":"section", "parent_id":"1", "category_id":null, "date":1375846025, "modified":1375846025, "attr":[] }, "gadgets":false } }</pre>
ERROR	<pre>{ "success":0, "message":"error message" }</pre>

Required Parameters (meta[] =)				
	Name	Type	Example	Notes
	meta[name]	STRING	TestLIR	Name of the new LIR resource.
	meta[type]	STRING	entry	Type of resource - Adding a LIR will always be "entry".
	meta[section]	STRING	lir	Section of the resource object. - Adding a LIR will always have "lir" as the section.

Fields Information

Detail on the "fields" array:

fields []	ARRAY	Entry field values (for fields that have already been assigned to the section) can be populated when the entry is created. The format is field[field-slug][field-instance]. If the field instance is left blank, it will simply be the next value in the instance array. For example: <i>fields[network-fqdn][]=example.com&fields[network-fqdn][]=test.com</i> would be written in JSON as <pre>var fields = { "network-fqdn": ["example.com", "test.com"] }</pre> Please note that the Field Slug might differ from the Field Name! To find the correct slug to use in adding resources with field values go the Section of the Resource you are adding, click 'Edit', then click the name of the Fields you will be populating. An Edit Field box will pop up which displays the Field's slug. A field can be added to a section multiple times. The field instance is used to keep track of which field occurrence we are referring. In this example, the network-fqdn field had been added twice to the section so we were able to store two values for it.
--------------	-------	---

Required Parameters

(fields[][] =>)

Name	Type	Example	Notes
fields[rir][]	STRING	ARIN	The RIR for the LIR. Accepted values are "ARIN", "RIPE", "LACNIC", "AfrinIC", "APNIC", and "1918".
fields[asn] []	STRING	20202	The ASN (Autonomous System Number) for the LIR.

Optional Parameters (fields[][] =)	For LIRs with RIR = "ARIN":			
	Name	Type	Example	Notes
	fields[org-handle][]	STRING	org1	Organization Handle. NOTE: When providing field information, Organization handle is required in order for the remaining fields to populate.
	fields[admin-poc][]	STRING	admin1	The admin point of contact object name.
	fields[net-poc][]	STRING	tech1	The network point of contact for this object.
	fields[abuse-poc][]	STRING	abuse1	The abuse point of contact object name.
	fields[net-name-prefix][]	STRING	test-	The network name prefix.
	fields[api-key][]	STRING	akakakakakaka	API key used to interface with RIR services
	For LIRs with RIR = "RIPE", "LACNIC", "AfriNIC", "APNIC", and "1918". :			
	Name	Type	Example	Notes
	fields[mnt-by][]	STRING	maint1	The name of the maintainer object. NOTE: When providing field information, Maintainer name is required in order for the remaining fields to populate.
	fields[mnt-by-password][]	STRING	passwordabcd	The maintainer password.
	fields[admin-c][]	STRING	admin1	Administrative contact for this object.
	fields[tech-c][]	STRING	tech1	Technical contact for this object
Example URL	ARIN example with only minimum required information:			
	/api/v1/api.php?target=lir&action=add&meta[name]=TestLIR1&meta[type]=entry&meta[section]=lir&fields[rir][]=ARIN&fields[asn][]=20202			
	ARIN example with full Organization field details:			
	/api/v1/api.php?target=lir&action=add&meta[name]=TestLIR2&meta[type]=entry&meta[section]=lir&fields[rir][]=ARIN&fields[asn][]=20202&fields[org-handle][]=org1&fields[admin-poc][]=admin1&fields[net-poc][]=tech1&fields[abuse-poc][]=abuse1&fields[net-name-prefix][]=test-&fields[api-key][]=akakakakakaka			
	Non-ARIN example with only minimum required information:			
	/api/v1/api.php?target=lir&action=add&meta[name]=TestLIR3&meta[type]=entry&meta[section]=lir&fields[rir][]=RIPE&fields[asn][]=20202			
	Non-ARIN example with full Organization field details:			
	/api/v1/api.php?target=lir&action=add&meta[name]=TestLIR4&meta[type]=entry&meta[section]=lir&fields[rir][]=1918&fields[asn][]=20202&fields[mnt-by][]=maint1&fields[mnt-by-password][]=passwordabcd&fields[admin-c][]=admin1&fields[tech-c][]=tech1			

Update

URL	/api/v1/api.php?target=lir&action=update
Description	Updates a LIR. LIR updates utilize the Resource system - see API Module - Resource - Update for details on editing a resource via the API.
Returns	Examples:

SUCCESSFUL	<pre>{ "success":1, "message":"Resource Updated", "data": { "id":"206", "name":"TestName", "slug":"dstl-reserved", "type":"entry", "parent_id":"191", "category_id":null, "date":1470755920, "modified":1517513903, "attr": { "mnt-by-password":11, "rir":"RIPE", "asn":"30303", "mnt-by":"Maint2", "admin-c":"admin4", "tech-c":"Tech5", "_section":"162", "_custom_id":"", "_gadget_data":[] }, "section": { "id":"162", "name":"4BCH", "slug":"4bch", "type":"section", "parent_id":"1", "category_id":null, "date":1468356788, "modified":1468356788, "attr":[] }, "gadgets":[] } }</pre>
------------	---

	<table><tr><td>ERROR</td><td><pre>{ "success":0, "message":"error message" }</pre></td></tr></table>	ERROR	<pre>{ "success":0, "message":"error message" }</pre>										
ERROR	<pre>{ "success":0, "message":"error message" }</pre>												
Required Parameters (meta[] =)	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Notes</th></tr><tr><td>meta[id]</td><td>INTEGER</td><td>1234</td><td>ID of the LIR resource to update.</td></tr><tr><td>meta[type]</td><td>STRING</td><td>entry</td><td>Type of resource - Updating a LIR will always be "entry".</td></tr></table>	Name	Type	Example	Notes	meta[id]	INTEGER	1234	ID of the LIR resource to update.	meta[type]	STRING	entry	Type of resource - Updating a LIR will always be "entry".
Name	Type	Example	Notes										
meta[id]	INTEGER	1234	ID of the LIR resource to update.										
meta[type]	STRING	entry	Type of resource - Updating a LIR will always be "entry".										
Optional Parameters (meta[] =)	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Notes</th></tr><tr><td>meta[name]</td><td>STRING</td><td>TestLIR</td><td>Name of the LIR resource.</td></tr></table>	Name	Type	Example	Notes	meta[name]	STRING	TestLIR	Name of the LIR resource.				
Name	Type	Example	Notes										
meta[name]	STRING	TestLIR	Name of the LIR resource.										
Required Parameters (fields[][] =)	None.												

Optional Parameters (fields[][] =)	For LIRs with RIR = "ARIN":			
	Name	Type	Example	Notes
	fields[rir][]	STRING	ARIN	The RIR for the LIR. Accepted values are "ARIN", "RIPE", "LACNIC", "AfriNIC", "APNIC", and "1918".
	fields[asn][]	STRING	20202	The ASN (Autonomous System Number) for the LIR.
	fields[org-handle][]	STRING	org1	Organization Handle. NOTE: When providing field information, Organization handle is required in order for the remaining Org. object fields to populate.
	fields[admin-poc][]	STRING	admin1	The admin point of contact object name.
	fields[net-poc][]	STRING	tech1	The network point of contact for this object.
	fields[abuse-poc][]	STRING	abuse1	The abuse point of contact object name.
	fields[net-name-prefix][]	STRING	test-	The network name prefix.
	fields[api-key][]	STRING	akakakakakaka	API key used to interface with RIR services
	For LIRs with RIR = "RIPE", "LACNIC", "AfriNIC", "APNIC", and "1918". :			
	Name	Type	Example	Notes
	fields[rir][]	STRING	RIPE	The RIR for the LIR. Accepted values are "ARIN", "RIPE", "LACNIC", "AfriNIC", "APNIC", and "1918".
	fields[asn][]	STRING	20202	The ASN (Autonomous System Number) for the LIR.
	fields[mnt-by][]	STRING	maint1	The name of the maintainer object. NOTE: When providing field information, Maintainer name is required in order for the remaining maintainer object fields to populate.
	fields[mnt-by-password][]	STRING	passwordabcd	The maintainer password.
	fields[admin-c][]	STRING	admin1	Administrative contact for this object.
	fields[tech-c][]	STRING	tech1	Technical contact for this object
Example URL	Update example with only the minimum required information:			
	This sends only the meta data required for a success response, but it does not actually change any values. /api/v1/api.php?target=lir&action=update&meta[id]=14068&meta[type]=entry Update example with changing RIR, ASN, and updating all field information: This example changes a non-ARIN LIR to an ARIN LIR, and provides all updated information for the new RIR type. /api/v1/api.php?target=lir&action=update&meta[id]=14068&meta[type]=entry&meta[name]=TestLIR4b&fields[rir][]=ARIN&fields[asn][]=30303&fields[org-handle][]=OrgABC&fields[admin-poc][]=AdminABC&fields[tech-c][]=TechABC&fields[abuse-poc][]=AbuseABC&fields[net-name-prefix][]=TestABC-&fields[api-key][]=abcdabcd			