

Get a list of dependencies for a card

GET /io/card/:cardId/dependency

Get a list of dependencies for a card

Query Params

Param	Type	Usage	Default
<code>includeFaces</code>	boolean	Pass "true" to also include the card face with each result	

Example Request

```
curl -X GET \
  https://myaccount.leankit.com/io/card/943206946/dependency?includeFaces=true \
  -H 'Authorization: Basic base64encodedauthhere' \
  -H 'Content-Type: application/json'
```

Note: dependency direction field can either be incoming or outgoing. An incoming dependency is one where the original card (from the url) depends on the card listed in the dependency. An outgoing dependency is one where the original card (from the url) is depended on by the card listed in the dependency.

Timing values:

- *startToStart* - Dependency can't start until the card starts
- *startToFinish* - Dependency can't finish until the card starts
- *finishToStart* - Dependency can't start until the card finishes
- *finishToFinish* - Dependency can't finish until the card finishes

Example Successful Response

200 OK

```

{
  "dependencies": [
    {
      "direction": "outgoing",
      "cardId": "5252812",
      "timing": "finishToFinish",
      "createdOn": "2025-06-10T19:32:57.237Z",
      "face": {
        "id": "5252812",
        "title": "3",
        "index": 2,
        "laneId": "5147995",
        "boardId": "5147970",
        "cardStatus": "started",
        "color": "#BA7070",
        "tags": [],
        "size": 0,
        "priority": "normal",
        "plannedStart": null,
        "plannedFinish": null,
        "actualStart": "2025-06-10T19:30:59Z",
        "actualFinish": null,
        "isDone": false,
        "movedOn": "2025-06-10T19:30:59.053Z",
        "updatedOn": "2025-06-10T19:40:51.150Z",
        "externalLinks": [],
        "customIconLabel": "Class of Service",
        "isMirroredCard": false,
        "blockedStatus": {
          "isBlocked": false,
          "reason": null
        },
        "customIcon": null,
        "customHeader": {
          "value": null,
          "header": null,
          "url": null
        },
        "taskBoardStats": null,
        "containingCardId": null,
        "cardType": {
          "id": "5147974",
          "name": "Architectural Enabler Story"
        },
        "subscriptionId": null,
        "attachmentsCount": 0,
        "commentsCount": 0,
        "archivedOn": null,
        "parentCards": [],
        "assignedUsers": [],
        "connectedCardStats": null,
        "customFields": {},
        "planningIncrements": [],
        "reactions": [],

```

```

{
  "assignedTeams": [],
  "policyRuleFieldIdViolations": []
}
```

Response Properties

Property	Type	Note
direction	string	outgoing (depends on the :cardId) or incoming (depended on by the :cardId)
cardId	integer id	internal unique id
timing	enumeration	Possible values: startToStart, startToFinish, finishToStart or finishToFinish
createdOn	date	The date that the dependency was created
face.id	integer id	internal unique id
face.title	string	
face.index	integer	The position of card (or task) in current lane.
face.laneId	integer	The internal unique id of current lane.
face.boardId	integer	The internal unique id of board.
face.color	hex value	
face.tags	string array	for example: ["bob", "sam"]
face.size	integer	The user-determined size of card (or task).
face.priority	priority value	"low", "normal", "high", or "critical"
face.plannedStart	date	
face.plannedFinish	date	

Property	Type	Note
<code>face.actualStart</code>	date	
<code>face.actualFinish</code>	date	
<code>face.archivedOn</code>	date	
<code>face.isDone</code>	boolean	Returns true when the card has been moved to archive or a done lane.
<code>face.movedOn</code>	date	
<code>face.updatedOn</code>	date	
<code>face.externalLinks</code>	externalLink array	{ label: "instagram", url: "http://instagram.com" }
<code>face.customIcon.id</code>	integer id	
<code>face.customIcon.cardColor</code>	hex value	
<code>face.customIcon.iconColor</code>	hex value	
<code>face.customIcon.iconName</code>	string	
<code>face.customIcon.iconPath</code>	string	
<code>face.blockedStatus.isBlocked</code>	boolean	
<code>face.blockedStatus.reason</code>	string	
<code>face.customHeader.value</code>	string	Depending on configuration, this may appear in the card or task header.
<code>face.customHeader.header</code>	string	The computed value of the card's header. It is the value prefixed with <code>customId.prefix</code> below.
<code>face.customHeader.url</code>	string	When configured, displays the url link for the header.
<code>face.taskBoardStats.totalCount</code>	integer	
<code>face.taskBoardStats.completedCount</code>	integer	
<code>face.taskBoardStats.totalSize</code>	integer	
<code>face.taskBoardStats.completedSize</code>	integer	

Property	Type	Note
<code>face.containingCardId</code>	integer	This is populated when the current object is a task.
<code>face.cardType.id</code>	integer	
<code>face.cardType.name</code>	string	
<code>face.subscriptionId</code>	integer	This is for internal subscription tracking only; do not use.
<code>scoring</code>	object	Card scoring data, Example: { confidenceTotal: 34, scoreOverride: 99, scoreTotal: 72, isTemplateChanged: false, scores: [{ meface.tricId: 1, score: 27.1, confidence: 45.7 }] }
<code>face.parentCards</code>	array of parentCard	Example: { id: 123, title: "A parent card" }
<code>face.assignedUsers</code>	array of users	Example: { id: 123, fullName: "John Smith", avatar: (link to avatar), emailAddress: "john@myco.com" }
<code>face.connectedCardStats.startedCount</code>	integer	
<code>face.connectedCardStats.startedSize</code>	integer	
<code>face.connectedCardStats.notStartedCount</code>	integer	
<code>face.connectedCardStats.notStartedSize</code>	integer	
<code>face.connectedCardStats.completedCount</code>	integer	
<code>face.connectedCardStats.completedSize</code>	integer	
<code>face.connectedCardStats.blockedCount</code>	integer	
<code>face.connectedCardStats.totalCount</code>	integer	
<code>face.connectedCardStats.totalSize</code>	integer	
<code>face.connectedCardStats.plannedStart</code>	date	

Property	Type	Note
<code>face.connectedCardStats.plannedFinish</code>	date	
<code>face.connectedCardStats.actualStart</code>	date	
<code>face.connectedCardStats.actualFinish</code>	date	
<code>face.connectedCardStats.pastDueCount</code>	integer	
<code>face.connectedCardStats.projectedLateCount</code>	integer	
<code>customFields</code>	object	The object keys are the custom field id, and the values are the custom field value. Example: { 943852689: "Custom Field Vaface.lue" }
<code>face.planningIncrementIds</code>	array of planning increment ids	
<code>face.dependencyStats.incomingCount</code>	integer	Incoming dependencies are cards that this card depends on
<code>face.dependencyStats.incomingResolvedCount</code>	integer	Incoming dependencies that are already resolved
<code>dependencyStats.incomingExceptionCount</code>	integer	Unresolved incoming dependencies that have exceptions (blocked, date mismatches, late to start or fiface.nish)
<code>face.dependencyStats.outgoingCount</code>	integer	Outgoing dependencies are cards that depend on this card
<code>face.dependencyStats.outgoingResolvedCount</code>	integer	Outgoing dependencies that are already resolved by this card
<code>dependencyStats.outgoingExceptionCount</code>	integer	Unresolved outgoing dependencies that have exceptions (blocked, date mismatches, late to start or fiface.nish)
<code>face.dependencyStats.totalCount</code>	integer	Total of incoming and outgoing dependencies
<code>face.dependencyStats.totalResolvedCount</code>	integer	Total resolved
<code>face.dependencyStats.totalExceptionCount</code>	integer	Total exceptions

Updated 2025-08-13 @ 12:08 PM