JSON File Data Reference

JSON File Data Reference

Example file for download: blueprint-definitions.json

Data Element	Structure	Description	Example

spaceBlueprints	Array	Lists every Brikit space blueprint on your site.
		
		Each item in this array is an object that defines one space blueprint.
		
		This array will be empty if there are no space blueprints for your site, as in the following:
		
		`"spaceBlueprints" : [`
		`]`
			If your site has two space blueprints, "New Product" and "Team Workspace":

spaceBlueprints[**i**]	Object	Defines an individual space blueprint on your site. This object comprises a set of key/value pairs that function as specifications or instructions for creating new spaces from this blueprint. Keys in this object: • name • description • blueprintSpaceKey • blueprintFields • spaceName • spaceKey • spaceCategories • spaceDescription • access	Object that defines a "New Product" space blueprint:
spaceBlueprints[i].**name**	Key/value pair	Defines the name of a space blueprint. The string value assigned to this key will be the name of the blueprint as it appears in the Blueprints Menu.	Naming a space blueprint, "New Product":
spaceBlueprints[i].**description**	Key/value pair	Describes a space blueprint. The string value assigned to this key will be the description of the blueprint as it appears directly below the blueprint's name/title in the Blueprints Menu.	Description for a "New Product" space blueprint:
spaceBlueprints[i].**blueprintSpaceKey**	Key/value pair	Indicates a blueprint source space. The string value assigned to this key is the space key of the source space that will serve as the template for new spaces generated from this blueprint.	Existing space whose key = "productspacebp" will be the template for this blueprint:

| spaceBlueprints[i].**blueprintFields** | Key/value pair | Lists all custom substitution fields included in the blueprint.<br><br>The array assigned to this key will comprise the names of all custom substitution fields that will be a part of this blueprint.<br><br>These are the fields blueprint users will be asked to provide values for whenever they create new spaces from this blueprint.<br><br>Each field name listed here will map to a blueprint field object defined in the JSON file. The substitution field name as it occurs in this array will match the corresponding blueprint field object's id key, which will also match the substitution field name as it occurs in the blueprint source page (enclosed in double brackets: [[substitution-field-name]]). | Substitution fields used in a "New Product" space blueprint:<br><br>```
"spaceBlueprints" : [
    ...
    {
        "name"               : "New Product",
        "description"        : "Create a space for a new product.",
        "blueprintSpaceKey"  : "productspacekey",
        "blueprintFields"    : [ "product-name", "product-owner-username" ],
        "spaceName"          : "[[product-name]]",
        "spaceKey"           : "[[product-name]]",
        "spaceCategories"    : "development, product",
        "spaceDescription"   : "Product space for everything related to [[product-name]].",
        "access"             : [ "Product Space" ]
    },
    ...
]
``` |
|---|---|---|---|
| spaceBlueprints[i].**spaceName** | Key/value pair | Defines how new spaces generated from a space blueprint will be named.<br><br>The string value of this key will be the name of a new space generated via blueprint.<br><br>It is possible and indeed recommended to incorporate at least one custom substitution field into this key's value so that each new space created from this blueprint will have a unique name from the start, without requiring any extra input from the user who is creating the new space. (Built-in substitution fields, however, are not supported here.) | How new spaces generated from a "Team Workspace" blueprint will be named:<br><br>```
"spaceBlueprints" : [
    ...
    {
        "name"               : "Team Workspace",
        "description"        : "Workspace for a department or project team",
        "blueprintSpaceKey"  : "teamspacekey",
        "blueprintFields"    : [ "team-name", "team-key", "team-edit-group" ],
        "spaceName"          : "[[team-name]] Workspace",
        "spaceKey"           : "[[team-key]]",
        "spaceDescription"   : "Projects, timelines, and resources for the [[team-name]] team",
        "access"             : [ "Team" ]
    },
    ...
]
``` |
| spaceBlueprints[i].**spaceKey** | Key/value pair | Defines the space key for new spaces generated from a blueprint.<br><br>The string value will be the space key of any new spaces generated via blueprint.<br><br>As with the spaceName key, it is possible and indeed recommended to incorporate at least one substitution field into this key's value so that each new space created from this blueprint will have a unique key from the start. Otherwise, when blueprint users try to create new spaces from blueprint, they will be prompted to go back and manually enter a unique space key to avoid naming collisions. | Naming convention for space keys generated from a "Team Workspace" blueprint:<br><br>```
"spaceBlueprints" : [
    ...
    {
        "name"               : "Team Workspace",
        "description"        : "Workspace for a department or project team",
        "blueprintSpaceKey"  : "teamspacekey",
        "blueprintFields"    : [ "team-name", "team-key", "team-edit-group" ],
        "spaceName"          : "[[team-name]] Workspace",
        "spaceKey"           : "[[team-key]]",
        "spaceDescription"   : "Projects, timelines, and resources for the [[team-name]] team",
        "access"             : [ "Team" ]
    },
    ...
]
``` |
| spaceBlueprints[i].**spaceCategories** | Key/value pair | Adds labels to new spaces generated from a blueprint.<br><br>The string value of this key is a comma-separated list of labels that will automatically be added to any new spaces created from the blueprint.<br><br>Custom substitution fields are allowed. The `[[CURRENT DATE]]`, `[[SPACE KEY]]`, and `[[SPACE NAME]]` built-in substitution fields are supported; however, we do not recommend using [[SPACE NAME]] as doing so will often result in errors (as labels cannot contain spaces and space names often consist of more than one word).<br><br>Leave blank or remove this key/value pair from the object altogether if you do not want to add labels to the new spaces. | Adding "development" and "product" labels to spaces created from this blueprint:<br><br>```
"spaceBlueprints" : [
    ...
    {
        "name"               : "New Product",
        "description"        : "Create a space for a new product.",
        "blueprintSpaceKey"  : "productspacekey",
        "blueprintFields"    : [ "product-name", "product-owner-username" ],
        "spaceName"          : "[[product-name]]",
        "spaceKey"           : "[[product-name]]",
        "spaceCategories"    : "development, product",
        "spaceDescription"   : "Product space for everything related to [[product-name]].",
        "access"             : [ "Product Space" ]
    },
    ...
]
``` |

spaceBlueprints[i].**spaceDescription**	Key/value pair	Describes new spaces generated via blueprint. The string value of this key will be the description (as seen in the Confluence space directory; editable via the Space Tools Menu) for any new space generated from the blueprint. It is possible to incorporate custom substitution fields into this key's value, should you wish to do that, although the use of built-in substitution fields is not supported.	Description metadata for spaces created from this blueprint: ```\n"spaceBlueprints" : [\n ...\n {\n "name" : "New Product",\n "description" : "Create a space for a new product.",\n "blueprintSpaceKey" : "productspacekey",\n "blueprintFields" : ["product-name", "product-owner-username"],\n "spaceName" : "{{product-name}}",\n "spaceKey" : "{{product-name}}",\n "spaceCategories" : "development, product",\n "spaceDescription" : "Product space for everything related to {{product-name}}",\n "access" : ["Product Space"]\n }\n ...\n]\n```
spaceBlueprints[i].**access**	Key/value pair	Assigns a permission scheme to new spaces created from a blueprint. The array assigned to this key will comprise the name of one or more permission settings objects enumerated in the accessLists array. Defining these objects will give blueprint authors tight control over who can view and edit the new spaces. Leave this array empty if you wish to restrict access to new spaces to everyone except the space creator.	Spaces created from this blueprint will inherit a "Product Space" permission scheme: ```\n"spaceBlueprints" : [\n ...\n {\n "name" : "New Product",\n "description" : "Create a space for a new product.",\n "blueprintSpaceKey" : "productspacekey",\n "blueprintFields" : ["product-name", "product-owner-username"],\n "spaceName" : "{{product-name}}",\n "spaceKey" : "{{product-name}}",\n "spaceCategories" : "development, product",\n "spaceDescription" : "Product space for everything related to {{product-name}}",\n "access" : ["Product Space"]\n }\n ...\n]\n```
accessLists	Array	Lists all permission schemes that may be applied to space blueprints. Each item in this array is an object that defines space permission settings. The objects in this list are what the items in spaceBlueprints[i].access may reference.	Defining "Product Space" and "Team" space permission settings: ```\n[unreadable code]\n```
accessLists[**m**]	Object	Defines a set of space permissions that may be applied to new spaces. This object comprises two key/value pairs that hold information about who can view, access, or administer spaces created from a blueprint. Keys in this object: - name - permissions	Object that defines a "Product Space" permission scheme: ```\n[unreadable code]\n```
accessLists[m].**name**	Key/value pair	Defines the name of a space permissions object whose properties can be applied to new spaces. The string value assigned to this key is simply the name of the containing permission settings object. Any instances of spaceBlueprints[i].access that reference this object will do so by this name.	Naming a space permission settings object, "Product Space": ```\n[unreadable code]\n```
accessLists[m].**permissions**	Key/value pair	Lists every type of user in the accessLists[m] permission scheme who will have some level of access to new spaces. Each item in this array is an object that defines specific permission settings for one particular user or group.	Setting permissions for three user groups and one individual project owner: ```\n[unreadable code]\n```

accessLists[m].permissions[**n**]	Object	Defines space permissions for a particular user or group. This object comprises a set of key/value pairs that hold information about how a particular user or group may interact new spaces created from any blueprint that references accessLists[m]. Keys in this object: • type • name • grant	Defining permission settings for a "developers" group: ```\n"accessLists" : [\n ...\n {\n "name" : "Product Space",\n "permissions" : [\n ...\n {\n "type" : "group",\n "name" : "developers",\n "grant" : ["VIEWSPACE", "EDITSPACE"]\n }\n ...\n]\n }\n ...\n]\n```

accessLists[m].permissions[n].**type**	Key/value pair	Specifies the type of user for whom space permissions are being set.
		
		The string value of this key will indicate whether accessLists[m].permissions[n] is defining space permissions for a Confluence user group, an individual user, or anonymous visitors.
		
		Valid values:
		
		• group
		• user
		• anonymous
			Specifying that permission settings are being defined for a user group:
			
			```\n"accessLists" : [\n ...\n {\n "name" : "Product Space",\n "permissions" : [\n ...\n {\n "type" : "group",\n "name" : "developers",\n "grant" : ["VIEWSPACE", "EDITSPACE"]\n }\n ...\n]\n }\n ...\n]\n```
			
			Specifying that permission settings are being defined for an individual user:
			
			```\n"accessLists" : [\n ...\n {\n "name" : "Product Space",\n "permissions" : [\n ...\n {\n "type" : "user",\n "name" : "[[product-owner-username]]",\n "grant" : ["ALL"]\n }\n ...\n]\n }\n ...\n]\n```

accessLists[m].permissions[n].**name**	Key/value pair	Specifies the name of the user or group for whom space permissions are being set.
		
		The string value of this key will be the name of a user group, an individual's username, or "anonymous", depending on the value of accessLists[m].permissions[n].type.
		
		It is possible to use custom substitution fields for this value, which blueprint users can supply at the time new spaces are created.
			Specifying that permission settings are being defined for a "developers" user group:
			
			```\n"accessLists" : [\n ...\n {\n "name" : "Product Space",\n "permissions" : [\n ...\n {\n "type" : "group",\n "name" : "developers",\n "grant" : ["VIEWSPACE", "EDITSPACE"]\n }\n ...\n]\n }\n ...\n]\n```
			
			Using a substitution field to set permissions for an individual:
			
			```\n"accessLists" : [\n ...\n {\n "name" : "Product Space",\n "permissions" : [\n ...\n {\n "type" : "user",\n "name" : "[[product-owner-username]]",\n "grant" : ["ALL"]\n }\n ...\n]\n }\n ...\n]\n```

accessLists[m].permissions[n].**grant**	Key/value pair	Lists all of the specific space permissions granted to the user or group defined in accessLists[m].permissions[n]. Each item in this array is a string value that corresponds with a specific type of access to a space an individual or group may possess. See valid space permissions array for a complete list of valid values.	Granting a "developers" group permissions to view and edit a space: ```\n"accessLists" : [\n ...\n {\n "name" : "Product Space",\n "permissions" : [\n ...\n {\n "type" : "group",\n "name" : "developers",\n "grant" : ["VIEWSPACE", "EDITSPACE"]\n }\n ...\n]\n }\n ...\n]\n```

valid space permissions	Array	Lists all of the different types of permissions that can be granted to a user or group in a new space created from a blueprint.
		
		For your convenience, all of the valid values you may use to populate accessLists[m].permissions[n].grant are listed here.
		
		We recommend keeping this array as it is and not removing any items so that you will always have a complete list to draw from:
		
		• ALL
		• VIEWSPACE
		• EDITSPACE
		• REMOVEPAGE
		• EDITBLOG
		• REMOVEBLOG
		• COMMENT
		• REMOVECOMMENT
		• CREATEATTACHMENT
		• REMOVEATTACHMENT
		• SETPAGEPERMISSIONS
		• EXPORTSPACE
		• SETSPACEPERMISSIONS
			All valid permissions that may be granted to users of new spaces:
			
			``
			"valid space permissions" : [
			    "ALL",
			    "VIEWSPACE",
			    "EDITSPACE",
			    "REMOVEPAGE",
			    "EDITBLOG",
			    "REMOVEBLOG",
			    "COMMENT",
			    "REMOVECOMMENT",
			    "CREATEATTACHMENT",
			    "REMOVEATTACHMENT",
			    "SETPAGEPERMISSIONS",
			    "EXPORTSPACE",
			    "SETSPACEPERMISSIONS"
			]
			``

pageBlueprints	Array	Lists every Brikit page blueprint on your site.
		
		Each item in this array is an object that defines one page blueprint .
		
		This array will be empty if there are no page blueprints for your site, as in the following:
		
		``
		"pageBlueprints" : [
		]
		``
			If your site has two page blueprints, "Product Overview" and "Team Member Profile":
			
			``
			"pageBlueprints" : [
			    ...
			    {
			        "name" : "Team Member Profile",
			        "description" : "Create a page to show a member's profile information.",
			        "spaceKey" : "BP",
			        "pageTitle" : "[[team-member-name]]",
			        "blueprintFields" : ["team-member-name", "team-member-username"],
			        "includeChildren" : true,
			        "visibility" : ["docs", "support"]
			    }
			    ...
			]
			``

pageBlueprints[j]	Object	Defines an individual page blueprint on your site.
		
		This object comprises a set of key/value pairs that function as specifications or instructions for creating new pages from this blueprint.
		
		Keys in this object:
		
		• name
		• description
		• spaceKey
		• pageTitle
		• blueprintFields
		• includeChildren *or* type
		• filterGroup
		• multiselect
		• addLabels
		• visibility
		
		Note : Blueprint Maker version 2.0 changed the **includeChildren** field to **type**
			Object that defines a "Team Member Profile" page blueprint:
			
			``
			"pageBlueprints" : [
			    ...
			    {
			        "name" : "Team Member Profile",
			        "description" : "Create a page to show a member's profile information.",
			        "spaceKey" : "BP",
			        "pageTitle" : "[[team-member-name]]",
			        "blueprintFields" : ["team-member-name", "team-member-username"],
			        "includeChildren" : true,
			        "visibility" : ["docs", "support"]
			    }
			    ...
			]
			``

pageBlueprints[j].**name**	Key/value pair	Defines the name of a page blueprint.
		
		The string value assigned to this key will be the name of the blueprint as it appears in the Blueprints Menu .
			Naming a page blueprint, "Product Overview":
			
			``
			"pageBlueprints" : [
			    ...
			    {
			        "name" : "Product Overview",
			        "description" : "Create a page that describes the product.",
			        "spaceKey" : "BP",
			        "pageTitle" : "[[product-name]] Overview",
			        "blueprintFields" : ["product-name", "product-logo"]
			    }
			    ...
			]
			``

pageBlueprints[j].**description**	Key/value pair	Describes a page blueprint.
		
		The string value assigned to this key will be the description of the blueprint as it appears directly below the blueprint's name/title in the Blueprints Menu.
			Description for a "Product Overview" page blueprint:
			
			``
			"pageBlueprints" : [
			    ...
			    {
			        "name" : "Product Overview",
			        "description" : "Create a page that describes the product.",
			        "spaceKey" : "BP",
			        "pageTitle" : "[[product-name]] Overview",
			        "blueprintFields" : ["product-name", "product-logo"]
			    }
			    ...
			]
			``

pageBlueprints[j].**spaceKey**	Key/value pair	Indicates the space that holds a page blueprint's source page. The string value will simply be the key of the space that is holding the blueprint source page. Another way to think about it: The space indicated here is where the value of pageBlueprints[j].**pageTitle** lives.	If the source page for this page blueprint lives in a space whose key = "BP": ```"pageBlueprints" : [` ` ...` ` {` ` "name" : "Product Overview",` ` "description" : "Create a page that describes the product.",` ` "spaceKey" : "BP",` ` "pageTitle" : "[[product-name]] Overview",` ` "blueprintFields" : ["product-name", "product-logo"]` ` }` ` ...` `]```
pageBlueprints[j].**pageTitle**	Key/value pair	Identifies a source page that will be used as the template for a blueprint. The string value assigned to this key will be the name of the source page that will be copied whenever new pages are generated from a blueprint. It is possible and indeed recommended to incorporate at least one custom substitution field into this key's value so that each new page created from this blueprint will have a unique name from the start, without requiring any extra input from the user who is creating the new space. Of the built-in substitution fields, `[[CURRENT DATE]]`, `[[SPACE KEY]]`, `[[PARENT PAGE ID]]` and `[[SPACE NAME]]` are supported.	Defining a page titled, "[[product-name]] Overview", as the source template page: ```"pageBlueprints" : [` ` ...` ` {` ` "name" : "Product Overview",` ` "description" : "Create a page that describes the product.",` ` "spaceKey" : "BP",` ` "pageTitle" : "[[product-name]] Overview",` ` "blueprintFields" : ["product-name", "product-logo"]` ` }` ` ...` `]```
pageBlueprints[j].**blueprintFields**	Key/value pair	Lists all custom substitution fields included in the blueprint. The array assigned to this key will comprise the names of all custom substitution fields that will be a part of this blueprint. These are the fields blueprint users will be asked to provide values for whenever they create new pages from this blueprint. Each field name listed here will map to a blueprint field object defined in the JSON file. The substitution field name as it occurs in this array will match the corresponding blueprint field object's id key, which will also match the substitution field name as it occurs in the blueprint source page (enclosed in double brackets: [[substitution-field-name]]).	Substitution fields used in a "Product Overview" page blueprint: ```"pageBlueprints" : [` ` ...` ` {` ` "name" : "Product Overview",` ` "description" : "Create a page that describes the product.",` ` "spaceKey" : "BP",` ` "pageTitle" : "[[product-name]] Overview",` ` "blueprintFields" : ["product-name", "product-logo"]` ` }` ` ...` `]```
pageBlueprints[j].**includeChildren**	Key/value pair	Indicates whether to include a blueprint source page's child pages when generating new pages from a blueprint. The boolean value assigned to this key will specify whether a source page's children will be a part of the blueprint, or just the source page indicated in pageBlueprint[j].pageTitle. Defaults to `false` if not indicated.	Child pages of "[[team-member-name]]" will be included in this blueprint: ```"pageBlueprints" : [` ` ...` ` {` ` "name" : "Team Member Profile",` ` "description" : "Create a page to show a member's profile information.",` ` "spaceKey" : "BP",` ` "pageTitle" : "[[team-member-name]]",` ` "blueprintFields" : ["team-member-name", "team-member-username"],` ` "includeChildren" : true,` ` "visibility" : ["docs", "support"]` ` }` ` ...` `]```
pageBlueprints[j]. **type**	Key/value pair	As of Blueprint Maker version 2.0, the **type** field replaces the **includeChildren** field. The **type** field indicates what kind of page Blueprint is to be created. If you simply want a single page created from a Blueprint (and no children) then enter `page`. If you would like to a page and its children (child pages) then enter `tree`. If you would like to create a Live Blueprint, where content can be refreshed after creation of the Blueprint, then enter `live`. Valid values: - `page` - `tree` - `live`	Child pages of "[[team-member-name]]" will be included in this blueprint: ```"pageBlueprints" : [` ` ...` ` {` ` "name" : "Team Member Profile",` ` "description" : "Create a page to show a member's profile information.",` ` "spaceKey" : "BP",` ` "pageTitle" : "[[team-member-name]]",` ` "blueprintFields" : ["team-member-name", "team-member-username"],` ` "type" : tree,` ` "visibility" : ["docs", "support"]` ` }` ` ...` `]```

| pageBlueprints[j].**visibility** | Array | This field lets users list in which spaces they want this Page Blueprint to be made available.<br><br>Add the **space category** label to this list if you want the Page Blueprint to be available in the menu of available Blueprints for use in a space.<br><br>Not including this field will mean the Page Blueprint is available to all spaces. | Visibility of this page Blueprint to space keys:<br><br>```
"pageBlueprints" : [
    ...
    {
        "name"            : "Team Member Profile",
        "description"     : "Create a page to show a member's profile information.",
        "spaceKey"        : "HF",
        "pageTitle"       : "[[team-member-name]]",
        "blueprintFields" : [ "team-member-name", "team-member-username" ],
        "includeChildren" : true,
        "visibility"      : [ "docs", "support" ]
    }
    ...
]
``` |
| **blueprintFields** | Array | Lists every substitution field used in your Blueprints.<br><br>Each item in this array is an object that defines one substitution field. | Define your substitution fields in the blueprintFields Array:<br><br>```
"blueprintFields" : [
    ...
    {
        "id"          : "team-member-name",
        "name"        : "Team Member Name",
        "required"    : true,
        "type"        : "medium",
        "helpText"    : "The full name of the team member.",
        "placeholder" : "Enter name here",
        "validation"  : "^[a-z-]+$"
    },
    ...
]
``` |
| blueprintFields[**k**] | Object | A substitution field is defined here as an object with properties.<br><br>A substitution field object comprises a set of key/value pairs that function as specifications or instructions for creating new spaces from this blueprint.<br><br>Keys in this object:<br><br>• id<br>• name<br>• required<br>• type<br>• filterGroup<br>• addLabels<br>• multiselect<br>• values<br>• helpText<br>• placeholder<br>• validation | Each substitution field is an object in the blueprintFields Array:<br><br>```
"blueprintFields" : [
    ...
    {
        "id"          : "team-member-name",
        "name"        : "Team Member Name",
        "required"    : true,
        "type"        : "medium",
        "helpText"    : "The full name of the team member.",
        "placeholder" : "Enter name here",
        "validation"  : "^[a-z-]+$"
    },
    ...
]
``` |
| blueprintFields[k].**id** | Key/value pair | The **id** field corresponds with the substitution field used on your Page and Space Blueprints. For example, if you used *[[team-member-name]]* on a Page Blueprint, then that is the **id** value used here, minus the brackets.<br><br>The **id** field is the unique identifier for your substitution field. It is typically a string of text, and always corresponds with the field used on your Page and Space Blueprints. | Each substitution field is a blueprintField Object:<br><br>```
"blueprintFields" : [
    ...
    {
                          "id"          : "team-member-name",
        "name"    : "Team Member Name",
        "required"    : true,
        "type"    : "medium",
        "helpText"    : "The full name of the team member.",
        "placeholder" : "Enter name here",
        "validation"  : "^[a-z-]+$"
    },
    ...
]
``` |
| blueprintFields[k].**name** | Key/value pair | The **name** field is a "Human readable" presentation of the substitution field. It is used in the Blueprint creation window to help users understand this field. | Give your substitution field a name:<br><br>```
"blueprintFields" : [
    ...
    {
        "id"          : "team-member-name",
                          "name"    : "Team Member Name",
        "required"    : true,
        "type"    : "medium",
        "helpText"    : "The full name of the team member.",
        "placeholder" : "Enter name here",
        "validation"  : "^[a-z-]+$"
    },
    ...
]
``` |
| blueprintFields[k].**required** | Boolean | The **required** field, if set to `true`, sets a flag for this substitution field that forces users to enter a value during the Blueprint creation process. This means a page or space cannot be created until the user enters a value for this field.<br><br>Possible values are:<br><br>• `true`<br>• `false` | Is your substitution field required?:<br><br>```
"blueprintFields" : [
    ...
    {
        "id"          : "team-member-name",
        "name"    : "Team Member Name",
                              "required"    : true,
        "type"    : "medium",
        "helpText"    : "The full name of the team member.",
        "placeholder" : "Enter name here",
        "validation"  : "^[a-z-]+$"
    },
    ...
]
``` |

blueprintFields[k].**type**	Key/value pair	The **type** field defines the field *length* or field *type* (if a file upload). The field length is useful for indicating to your users if there is an expectation of a short value being entered, such as a single word name, or a long value, such as a sentence or two.
		
		If desiring a file upload, set the type to `file` and the wizard will allow users to attach a file during the Blueprint creation process.
		
		For a picklist of options , enter `list` and then use either the `filterGroup` or `values` field.
		
		Valid values:
		
		• `default`
		• `short`
		• `medium`
		• `long`
		• `file`
		• `list`
			Give your substitution field a type:
			
			``
			"blueprintFields" : [
			    ...
			    {
			        "id" : "team-member-name",
			        "name" : "Team Member Name",
			        "required" : true,
			
			                      "type" : "medium",
			        "helpText" : "The full name of the team member.",
			        "placeholder" : "Enter name here",
			        "validation" : "^[a-z-]+$"
			    }, ...
			]
			``

blueprintFields[k]. **filterGroup**	Key/value pair	The **filterGroup** field is used with the "list" `type` to identify a Brikit Targeted Search Filter Group. The Filters in this Filter Group are used as the picklist options, and can be added as labels too, with `addLabels` set to true. Read more about using Targeted Search Filters with Blueprint Maker.	Enter the exact (case-sensitive) name of the Filter Group: ``` (code block) ```
blueprintFields[k]. **addLabels**	Boolean	When you set your `type` to "list" and **addLabels** to true, then the options in your `values` array will be used in the picklist and applied to the resulting page as labels. Read more about Applying Labels via Blueprint. Possible values are: • `true` • `false`	Want the picklist options to be applied as labels to the page? ``` (code block) ```
blueprintFields[k]. **multiselect**	Boolean	If your blueprintField `type` is set to "list" to use a list of `values` (or `filterGroup`) as an array, then set **multiselect** to true to allow users to select more than one option in the picklist. Possible values are: • `true` • `false`	Want users to select multiple options? ``` (code block) ```
blueprintFields[k]. **values**	Array	With a `type` set to "list", then define the picklist options in an array of **values** here. This is a comma-delimited set of text, each enclosed in quotation marks. **Note 1:** if `addLabels` is true, then the options in the `values` array must be valid Confluence labels. (if not, an error will occur). **Note 2**: if filterGroup and values are both defined, the options will use the filterGroup Filters (in Targeted Search) and ignore the values options.	Define a list of options to present to users in the picklist: ``` (code block) ```

blueprintFields[k].**helpText**	Key/value pair	The **helpText** field is an optional, short description or instruction that appears below the field in the Blueprint wizard. This is meant to give users more information about the field, and the kind of values they should be entering.	Give your substitution field some help text:
			
			``
			"blueprintFields" : [
			    ...
			    {
			        "id" : "team-member-name",
			        "name" : "Team Member Name",
			        "required" : true,
			        "type" : "medium",
			
			                      "helpText" : "The full name of the team member.",
			        "placeholder" : "Enter name here",
			        "validation" : "^[a-z-]+$"
			    }, ...
			]
			``

blueprintFields[k].**placeholder**	Key/value pair	The value of the **placeholder** field is presented in the field itself as a kind of sample text or succinct instruction to the user. Like the helpText, it is optional but good practice for assisting users in the Blueprint creation process.	Give your substitution field a placeholder value:
			
			``json
			"blueprintFields" : [
			    ...
			    {
			        "id" : "team-member-name",
			        "name" : "Team Member Name",
			        "required" : true,
			        "type" : "medium",
			        "helpText" : "The full name of the team member.",
			        "placeholder" : "Enter name here",
			        "validation" : "^[a-z-]+$"
			    },
			    ...
			]
			``

blueprintFields[k].**validation**	Key/value pair	The **validation** field take a regular expression and uses it to check that a user's input is valid. This is particularly useful to verify that a user has entered the expected kind of characters, such as *only text*, or *only numbers*, or a standard character pattern (as in a phone number or email address).
		
		Learn more about regular expressions .
			Validate your substitution field value using regex:
			
			``json
			"blueprintFields" : [
			    ...
			    {
			        "id" : "team-member-name",
			        "name" : "Team Member Name",
			        "required" : true,
			        "type" : "medium",
			        "helpText" : "The full name of the team member.",
			        "placeholder" : "Enter name here",
			        "validation" : "^[a-z-]+$"
			    },
			    ...
			]
			``

spaceMetadata	Array	A list of the Confluence and app metadata packages to be included during the creation of a Space Blueprint.
		
		We recommend keeping this array as it is and not removing any items in order to ensure successful Blueprint creation. The default list items are:
		
		- `atlassian.confluence.*`
		- `com.brikit.*`
			If the source page for this page blueprint lives in a space whose key = "BP":
			
			``json
			"spaceMetadata" : [
			    "atlassian.confluence.*",
			    "com.brikit.*"
			]
			``

pageMetadata	Array	A list of the app metadata packages to be included during the creation of a Page Blueprint.
		
		We recommend keeping this array as it is and not removing any items in order to ensure successful Blueprint creation. The default list item is:
		
		- `com.brikit.*`
			If the source page for this page blueprint lives in a space whose key = "BP":
			
			``json
			"pageMetadata" : [
			    "com.brikit.*"
			]
			``

createSpaceGroups	Array	A list of the groups who are allowed to create a Space Blueprint.
		
		BLUEPRINT-112
			If the source page for this page blueprint lives in a space whose key = "BP":
			
			``json
			"createSpaceGroups" : [
			    "blueprint-makers"
			]
			``

Related

- Built-In Substitution Fields
- Installing Blueprint Maker
- Blueprint Maker Technical Reference