

# Interfaces

---

## In this article

[About interfaces](#)

[Actor](#)

[Assignable](#)

[AuditEntry](#)

[Closable](#)

[Comment](#)

[Contribution](#)

[Deletable](#)

[EnterpriseAuditEntryData](#)

[GitObject](#)

[GitSignature](#)

[HovercardContext](#)

[Labelable](#)

[Lockable](#)

[MemberStatusable](#)

[Minimizable](#)

[Node](#)

[OauthApplicationAuditEntryData](#)

[OrganizationAuditEntryData](#)

[PackageOwner](#)

[ProfileOwner](#)

[ProjectOwner](#)

[Reactable](#)

[RepositoryAuditEntryData](#)

[RepositoryInfo](#)

[RepositoryNode](#)

[RepositoryOwner](#)

[Sponsorable](#)

Starrable  
Subscribable  
TeamAuditEntryData  
TopicAuditEntryData  
UniformResourceLocatable  
Updatable  
UpdatableComment

---

## About interfaces

**Interfaces** serve as parent objects from which other objects can inherit.

For example, `Lockable` is an interface because both `Issue` and `PullRequest` objects can be locked. An interface has its own list of named fields that are shared by implementing objects.

For more information, see "[Implementation](#)."

## Actor

Represents an object which can take actions on GitHub. Typically a User or Bot.

### Implemented by

- `Bot`
- `EnterpriseUserAccount`
- `Mannequin`
- `Organization`
- `User`

### Fields

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

---

Name	Description
avatarUrl ( <a href="#">URI!</a> )	A URL pointing to the actor's public avatar.
<div> <b>Arguments</b> </div> <div> size (<a href="#">Int</a>)  The size of the resulting square image. </div>	
login ( <a href="#">String!</a> )	The username of the actor.
resourcePath ( <a href="#">URI!</a> )	The HTTP path for this actor.
url ( <a href="#">URI!</a> )	The HTTP URL for this actor.

## Assignable

An object that can have users assigned to it.

### Implemented by

- [Issue](#)
- [PullRequest](#)

### Fields

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

Name	Description
<code>assignees(<a href="#">UserConnection!</a>)</code>	A list of Users assigned to this object.

#### Arguments

`after` ([String](#))

Returns the elements in the list that come after the specified cursor.

`before` ([String](#))

Returns the elements in the list that come before the specified cursor.

`first` ([Int](#))

Returns the first  $n$  elements from the list.

`last` ([Int](#))

Returns the last  $n$  elements from the list.

## AuditEntry

An entry in the audit log.

### Implemented by

- [MembersCanDeleteReposClearAuditEntry](#)
- [MembersCanDeleteReposDisableAuditEntry](#)
- [MembersCanDeleteReposEnableAuditEntry](#)
- [OauthApplicationCreateAuditEntry](#)
- [OrgAddBillingManagerAuditEntry](#)
- [OrgAddMemberAuditEntry](#)
- [OrgBlockUserAuditEntry](#)
- [OrgConfigDisableCollaboratorsOnlyAuditEntry](#)
- [OrgConfigEnableCollaboratorsOnlyAuditEntry](#)
- [OrgCreateAuditEntry](#)
- [OrgDisableOauthAppRestrictionsAuditEntry](#)

- `OrgDisableSamlAuditEntry`
- `OrgDisableTwoFactorRequirementAuditEntry`
- `OrgEnableOAuthAppRestrictionsAuditEntry`
- `OrgEnableSamlAuditEntry`
- `OrgEnableTwoFactorRequirementAuditEntry`
- `OrgInviteMemberAuditEntry`
- `OrgInviteToBusinessAuditEntry`
- `OrgOAuthAppAccessApprovedAuditEntry`
- `OrgOAuthAppAccessDeniedAuditEntry`
- `OrgOAuthAppAccessRequestedAuditEntry`
- `OrgRemoveBillingManagerAuditEntry`
- `OrgRemoveMemberAuditEntry`
- `OrgRemoveOutsideCollaboratorAuditEntry`
- `OrgRestoreMemberAuditEntry`
- `OrgUnblockUserAuditEntry`
- `OrgUpdateDefaultRepositoryPermissionAuditEntry`
- `OrgUpdateMemberAuditEntry`
- `OrgUpdateMemberRepositoryCreationPermissionAuditEntry`
- `OrgUpdateMemberRepositoryInvitationPermissionAuditEntry`
- `PrivateRepositoryForkingDisableAuditEntry`
- `PrivateRepositoryForkingEnableAuditEntry`
- `RepoAccessAuditEntry`
- `RepoAddMemberAuditEntry`
- `RepoAddTopicAuditEntry`
- `RepoArchivedAuditEntry`
- `RepoChangeMergeSettingAuditEntry`
- `RepoConfigDisableAnonymousGitAccessAuditEntry`
- `RepoConfigDisableCollaboratorsOnlyAuditEntry`
- `RepoConfigDisableContributorsOnlyAuditEntry`
- `RepoConfigDisableSockpuppetDisallowedAuditEntry`
- `RepoConfigEnableAnonymousGitAccessAuditEntry`
- `RepoConfigEnableCollaboratorsOnlyAuditEntry`
- `RepoConfigEnableContributorsOnlyAuditEntry`
- `RepoConfigEnableSockpuppetDisallowedAuditEntry`
- `RepoConfigLockAnonymousGitAccessAuditEntry`
- `RepoConfigUnlockAnonymousGitAccessAuditEntry`

- `RepoCreateAuditEntry`
- `RepoDestroyAuditEntry`
- `RepoRemoveMemberAuditEntry`
- `RepoRemoveTopicAuditEntry`
- `RepositoryVisibilityChangeDisableAuditEntry`
- `RepositoryVisibilityChangeEnableAuditEntry`
- `TeamAddMemberAuditEntry`
- `TeamAddRepositoryAuditEntry`
- `TeamChangeParentTeamAuditEntry`
- `TeamRemoveMemberAuditEntry`
- `TeamRemoveRepositoryAuditEntry`

## Fields

Name	Description
<code>action</code> ( <code>String!</code> )	The action name.
<code>actor</code> ( <code>AuditEntryActor</code> )	The user who initiated the action.
<code>actorIp</code> ( <code>String</code> )	The IP address of the actor.
<code>actorLocation</code> ( <code>ActorLocation</code> )	A readable representation of the actor's location.
<code>actorLogin</code> ( <code>String</code> )	The username of the user who initiated the action.
<code>actorResourcePath</code> ( <code>URI</code> )	The HTTP path for the actor.
<code>actorUrl</code> ( <code>URI</code> )	The HTTP URL for the actor.
<code>createdAt</code> ( <code>PreciseDateTime!</code> )	The time the action was initiated.
<code>operationType</code> ( <code>OperationType</code> )	The corresponding operation type for the action.
<code>user</code> ( <code>User</code> )	The user affected by the action.
<code>userLogin</code> ( <code>String</code> )	For actions involving two users, the actor is the initiator and the user is the affected user.

Name	Description
<code>userResourcePath(<a href="#">URI</a>)</code>	The HTTP path for the user.
<code>userUrl (<a href="#">URI</a>)</code>	The HTTP URL for the user.

---

## Closable

An object that can be closed.

### Implemented by

- [Issue](#)
- [Milestone](#)
- [Project](#)
- [PullRequest](#)

### Fields

Name	Description
<code>closed (<a href="#">Boolean!</a>)</code>	true if the object is closed (definition of closed may depend on type).
<code>closedAt (<a href="#">DateTime</a>)</code>	Identifies the date and time when the object was closed.

---

## Comment

Represents a comment.

### Implemented by

- [CommitComment](#)
- [GistComment](#)

- `Issue`
- `IssueComment`
- `PullRequest`
- `PullRequestReview`
- `PullRequestReviewComment`
- `TeamDiscussion`
- `TeamDiscussionComment`

## Fields

Name	Description
<code>author</code> ( <code>Actor</code> )	The actor who authored the comment.
<code>authorAssociation</code> ( <code>CommentAuthorAssociation!</code> )	Author's association with the subject of the comment.
<code>body</code> ( <code>String!</code> )	The body as Markdown.
<code>bodyHTML</code> ( <code>HTML!</code> )	The body rendered to HTML.
<code>bodyText</code> ( <code>String!</code> )	The body rendered to text.
<code>createdAt</code> ( <code>DateTime!</code> )	Identifies the date and time when the object was created.
<code>createdViaEmail</code> ( <code>Boolean!</code> )	Check if this comment was created via an email reply.
<code>editor</code> ( <code>Actor</code> )	The actor who edited the comment.
<code>includesCreatedEdit</code> ( <code>Boolean!</code> )	Check if this comment was edited and includes an edit with the creation data.
<code>lastEditedAt</code> ( <code>DateTime</code> )	The moment the editor made the last edit.
<code>publishedAt</code> ( <code>DateTime</code> )	Identifies when the comment was published at.
<code>updatedAt</code> ( <code>DateTime!</code> )	Identifies the date and time when the object was last updated.



Name	Description
<code>userContentEdits</code> ( <code>UserContentEditConnection</code> )	<p>A list of edits to this content.</p> <div> <p><b>Arguments</b></p> <hr/> <p><code>after</code> (<code>String</code>) Returns the elements in the list that come after the specified cursor.</p> <hr/> <p><code>before</code> (<code>String</code>) Returns the elements in the list that come before the specified cursor.</p> <hr/> <p><code>first</code> (<code>Int</code>) Returns the first <math>n</math> elements from the list.</p> <hr/> <p><code>last</code> (<code>Int</code>) Returns the last <math>n</math> elements from the list.</p> </div>
<code>viewerDidAuthor</code> ( <code>Boolean!</code> )	Did the viewer author this comment.

## Contribution

Represents a contribution a user made on GitHub, such as opening an issue.

### Implemented by

- `CreatedCommitContribution`
- `CreatedIssueContribution`
- `CreatedPullRequestContribution`
- `CreatedPullRequestReviewContribution`
- `CreatedRepositoryContribution`
- `JoinedGitHubContribution`
- `RestrictedContribution`

### Fields

Name	Description
<code>isRestricted(Boolean!)</code>	Whether this contribution is associated with a record you do not have access to. For example, your own 'first issue' contribution may have been made on a repository you can no longer access.
<code>occurredAt(DateTime!)</code>	When this contribution was made.
<code>resourcePath(URI!)</code>	The HTTP path for this contribution.
<code>url (URI!)</code>	The HTTP URL for this contribution.
<code>user (User!)</code>	The user who made this contribution.

## Deletable

Entities that can be deleted.

### Implemented by

- `CommitComment`
- `GistComment`
- `IssueComment`
- `PullRequestReview`
- `PullRequestReviewComment`
- `TeamDiscussion`
- `TeamDiscussionComment`

### Fields

Name	Description
<code>viewerCanDelete (Boolean!)</code>	Check if the current viewer can delete this object.

## EnterpriseAuditEntryData

Metadata for an audit entry containing enterprise account information.

### Implemented by

- `MembersCanDeleteReposClearAuditEntry`
- `MembersCanDeleteReposDisableAuditEntry`
- `MembersCanDeleteReposEnableAuditEntry`
- `OrgInviteToBusinessAuditEntry`
- `PrivateRepositoryForkingDisableAuditEntry`
- `PrivateRepositoryForkingEnableAuditEntry`
- `RepositoryVisibilityChangeDisableAuditEntry`
- `RepositoryVisibilityChangeEnableAuditEntry`

### Fields

Name	Description
<code>enterpriseResourcePath</code> ( <a href="#">URI</a> )	The HTTP path for this enterprise.
<code>enterpriseSlug</code> ( <a href="#">String</a> )	The slug of the enterprise.
<code>enterpriseUrl</code> ( <a href="#">URI</a> )	The HTTP URL for this enterprise.

## GitObject

Represents a Git object.

### Implemented by

- `Blob`
- `Commit`
- `Tag`
- `Tree`

## Fields

Name	Description
abbreviatedOid ( <a href="#">String!</a> )	An abbreviated version of the Git object ID.
commitResourcePath ( <a href="#">URI!</a> )	The HTTP path for this Git object.
commitUrl ( <a href="#">URI!</a> )	The HTTP URL for this Git object.
oid ( <a href="#">GitObjectID!</a> )	The Git object ID.
repository ( <a href="#">Repository!</a> )	The Repository the Git object belongs to.

## GitSignature

Information about a signature (GPG or S/MIME) on a Commit or Tag.

### Implemented by

- [GpgSignature](#)
- [SmimeSignature](#)
- [UnknownSignature](#)

## Fields

Name	Description
email ( <a href="#">String!</a> )	Email used to sign this object.
isValid ( <a href="#">Boolean!</a> )	True if the signature is valid and verified by GitHub.
payload ( <a href="#">String!</a> )	Payload for GPG signing object. Raw ODB object without the signature header.
signature( <a href="#">String!</a> )	ASCII-armored signature header from object.

Name	Description
signer ( <a href="#">User</a> )	GitHub user corresponding to the email signing this commit.
state ( <a href="#">GitSignatureState!</a> )	The state of this signature. <code>VALID</code> if signature is valid and verified by GitHub, otherwise represents reason why signature is considered invalid.
wasSignedByGitHub( <a href="#">Boolean!</a> )	True if the signature was made with GitHub's signing key.

## HovercardContext

An individual line of a hovercard.

### Implemented by

- [GenericHovercardContext](#)
- [OrganizationTeamsHovercardContext](#)
- [OrganizationsHovercardContext](#)
- [ReviewStatusHovercardContext](#)
- [ViewerHovercardContext](#)

### Fields

Name	Description
message ( <a href="#">String!</a> )	A string describing this context.
octicon ( <a href="#">String!</a> )	An octicon to accompany this context.

## Labelable

An object that can have labels assigned to it.

## Implemented by

- `Issue`
- `PullRequest`

## Fields

Name

Description

---

`labels`(`LabelConnection`) A list of labels associated with the object.

### Arguments

`after` (`String`)

Returns the elements in the list that come after the specified cursor.

`before` (`String`)

Returns the elements in the list that come before the specified cursor.

`first` (`Int`)

Returns the first  $n$  elements from the list.

`last` (`Int`)

Returns the last  $n$  elements from the list.

`orderBy` (`LabelOrder`)

Ordering options for labels returned from the connection.

---

## Lockable

An object that can be locked.

## Implemented by

- `Issue`
- `PullRequest`

## Fields

Name	Description
activeLockReason ( <a href="#">LockReason</a> )	Reason that the conversation was locked.
locked ( <a href="#">Boolean!</a> )	true if the object is locked.

---

## MemberStatusable

Entities that have members who can set status messages.

### Implemented by

- [Organization](#)
- [Team](#)

## Fields

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

---

Name	Description
memberStatuses ( <a href="#">UserStatusConnection!</a> )	Get the status messages members of this entity have set that are either public or visible only to the organization.
<div> <div>Arguments</div> <div> <div>after (<a href="#">String</a>) Returns the elements in the list that come after the specified cursor.</div> <div>before (<a href="#">String</a>) Returns the elements in the list that come before the specified cursor.</div> <div>first (<a href="#">Int</a>) Returns the first <math>n</math> elements from the list.</div> <div>last (<a href="#">Int</a>) Returns the last <math>n</math> elements from the list.</div> <div>orderBy (<a href="#">UserStatusOrder</a>) Ordering options for user statuses returned from the connection.</div> </div> </div>	

## Minimizable

Entities that can be minimized.

### Implemented by

- [CommitComment](#)
- [GistComment](#)
- [IssueComment](#)
- [PullRequestReviewComment](#)

### Fields



Name	Description
isMinimized ( <a href="#">Boolean!</a> )	Returns whether or not a comment has been minimized.
minimizedReason ( <a href="#">String</a> )	Returns why the comment was minimized.
viewerCanMinimize ( <a href="#">Boolean!</a> )	Check if the current viewer can minimize this object.

## Node

An object with an ID.

### Implemented by

- [AddedToProjectEvent](#)
- [App](#)
- [AssignedEvent](#)
- [AutomaticBaseChangeFailedEvent](#)
- [AutomaticBaseChangeSucceededEvent](#)
- [BaseRefChangedEvent](#)
- [BaseRefDeletedEvent](#)
- [BaseRefForcePushedEvent](#)
- [Blob](#)
- [Bot](#)
- [BranchProtectionRule](#)
- [CheckRun](#)
- [CheckSuite](#)
- [ClosedEvent](#)
- [CodeOfConduct](#)
- [CommentDeletedEvent](#)
- [Commit](#)
- [CommitComment](#)
- [CommitCommentThread](#)
- [ConnectedEvent](#)
- [ConvertToDraftEvent](#)

- `ConvertedNoteToIssueEvent`
- `CrossReferencedEvent`
- `DemilestonedEvent`
- `DependencyGraphManifest`
- `DeployKey`
- `DeployedEvent`
- `Deployment`
- `DeploymentEnvironmentChangedEvent`
- `DeploymentStatus`
- `DisconnectedEvent`
- `Enterprise`
- `EnterpriseAdministratorInvitation`
- `EnterpriseIdentityProvider`
- `EnterpriseRepositoryInfo`
- `EnterpriseServerInstallation`
- `EnterpriseServerUserAccount`
- `EnterpriseServerUserAccountEmail`
- `EnterpriseServerUserAccountsUpload`
- `EnterpriseUserAccount`
- `ExternalIdentity`
- `Gist`
- `GistComment`
- `HeadRefDeletedEvent`
- `HeadRefForcePushedEvent`
- `HeadRefRestoredEvent`
- `IpAllowListEntry`
- `Issue`
- `IssueComment`
- `Label`
- `LabeledEvent`
- `Language`
- `License`
- `LockedEvent`
- `Mannequin`
- `MarkedAsDuplicateEvent`
- `MarketplaceCategory`

- MarketplaceListing
- MembersCanDeleteReposClearAuditEntry
- MembersCanDeleteReposDisableAuditEntry
- MembersCanDeleteReposEnableAuditEntry
- MentionedEvent
- MergedEvent
- Milestone
- MilestonedEvent
- MovedColumnsInProjectEvent
- OAuthApplicationCreateAuditEntry
- OrgAddBillingManagerAuditEntry
- OrgAddMemberAuditEntry
- OrgBlockUserAuditEntry
- OrgConfigDisableCollaboratorsOnlyAuditEntry
- OrgConfigEnableCollaboratorsOnlyAuditEntry
- OrgCreateAuditEntry
- OrgDisableOAuthAppRestrictionsAuditEntry
- OrgDisableSamlAuditEntry
- OrgDisableTwoFactorRequirementAuditEntry
- OrgEnableOAuthAppRestrictionsAuditEntry
- OrgEnableSamlAuditEntry
- OrgEnableTwoFactorRequirementAuditEntry
- OrgInviteMemberAuditEntry
- OrgInviteToBusinessAuditEntry
- OrgOAuthAppAccessApprovedAuditEntry
- OrgOAuthAppAccessDeniedAuditEntry
- OrgOAuthAppAccessRequestedAuditEntry
- OrgRemoveBillingManagerAuditEntry
- OrgRemoveMemberAuditEntry
- OrgRemoveOutsideCollaboratorAuditEntry
- OrgRestoreMemberAuditEntry
- OrgUnblockUserAuditEntry
- OrgUpdateDefaultRepositoryPermissionAuditEntry
- OrgUpdateMemberAuditEntry
- OrgUpdateMemberRepositoryCreationPermissionAuditEntry
- OrgUpdateMemberRepositoryInvitationPermissionAuditEntry

- `Organization`
- `OrganizationIdentityProvider`
- `OrganizationInvitation`
- `Package`
- `PackageFile`
- `PackageTag`
- `PackageVersion`
- `PinnedEvent`
- `PinnedIssue`
- `PrivateRepositoryForkingDisableAuditEntry`
- `PrivateRepositoryForkingEnableAuditEntry`
- `Project`
- `ProjectCard`
- `ProjectColumn`
- `PublicKey`
- `PullRequest`
- `PullRequestCommit`
- `PullRequestCommitCommentThread`
- `PullRequestReview`
- `PullRequestReviewComment`
- `PullRequestReviewThread`
- `Push`
- `PushAllowance`
- `Reaction`
- `ReadyForReviewEvent`
- `Ref`
- `ReferencedEvent`
- `Release`
- `ReleaseAsset`
- `RemovedFromProjectEvent`
- `RenamedTitleEvent`
- `ReopenedEvent`
- `RepoAccessAuditEntry`
- `RepoAddMemberAuditEntry`
- `RepoAddTopicAuditEntry`
- `RepoArchivedAuditEntry`

- RepoChangeMergeSettingAuditEntry
- RepoConfigDisableAnonymousGitAccessAuditEntry
- RepoConfigDisableCollaboratorsOnlyAuditEntry
- RepoConfigDisableContributorsOnlyAuditEntry
- RepoConfigDisableSockpuppetDisallowedAuditEntry
- RepoConfigEnableAnonymousGitAccessAuditEntry
- RepoConfigEnableCollaboratorsOnlyAuditEntry
- RepoConfigEnableContributorsOnlyAuditEntry
- RepoConfigEnableSockpuppetDisallowedAuditEntry
- RepoConfigLockAnonymousGitAccessAuditEntry
- RepoConfigUnlockAnonymousGitAccessAuditEntry
- RepoCreateAuditEntry
- RepoDestroyAuditEntry
- RepoRemoveMemberAuditEntry
- RepoRemoveTopicAuditEntry
- Repository
- RepositoryInvitation
- RepositoryTopic
- RepositoryVisibilityChangeDisableAuditEntry
- RepositoryVisibilityChangeEnableAuditEntry
- RepositoryVulnerabilityAlert
- ReviewDismissalAllowance
- ReviewDismissedEvent
- ReviewRequest
- ReviewRequestRemovedEvent
- ReviewRequestedEvent
- SavedReply
- SecurityAdvisory
- SponsorsListing
- SponsorsTier
- Sponsorship
- Status
- StatusCheckRollup
- StatusContext
- SubscribedEvent
- Tag

- `Team`
- `TeamAddMemberAuditEntry`
- `TeamAddRepositoryAuditEntry`
- `TeamChangeParentTeamAuditEntry`
- `TeamDiscussion`
- `TeamDiscussionComment`
- `TeamRemoveMemberAuditEntry`
- `TeamRemoveRepositoryAuditEntry`
- `Topic`
- `TransferredEvent`
- `Tree`
- `UnassignedEvent`
- `UnlabeledEvent`
- `UnlockedEvent`
- `UnmarkedAsDuplicateEvent`
- `UnpinnedEvent`
- `UnsubscribedEvent`
- `User`
- `UserBlockedEvent`
- `UserContentEdit`
- `UserStatus`

### Fields

Name	Description
<code>id</code> ( <code>ID!</code> )	ID of the object.

## OauthApplicationAuditEntryData

Metadata for an audit entry with action `oauth_application.*`.

### Implemented by

- `OauthApplicationCreateAuditEntry`

- `OrgOAuthAppAccessApprovedAuditEntry`
- `OrgOAuthAppAccessDeniedAuditEntry`
- `OrgOAuthAppAccessRequestedAuditEntry`

## Fields

Name	Description
<code>oauthApplicationName</code> ( <a href="#">String</a> )	The name of the OAuth Application.
<code>oauthApplicationResourcePath</code> ( <a href="#">URI</a> )	The HTTP path for the OAuth Application.
<code>oauthApplicationUrl</code> ( <a href="#">URI</a> )	The HTTP URL for the OAuth Application.

## OrganizationAuditEntryData

Metadata for an audit entry with action org.\*.

### Implemented by

- `MembersCanDeleteReposClearAuditEntry`
- `MembersCanDeleteReposDisableAuditEntry`
- `MembersCanDeleteReposEnableAuditEntry`
- `OAuthApplicationCreateAuditEntry`
- `OrgAddBillingManagerAuditEntry`
- `OrgAddMemberAuditEntry`
- `OrgBlockUserAuditEntry`
- `OrgConfigDisableCollaboratorsOnlyAuditEntry`
- `OrgConfigEnableCollaboratorsOnlyAuditEntry`
- `OrgCreateAuditEntry`
- `OrgDisableOAuthAppRestrictionsAuditEntry`
- `OrgDisableSamlAuditEntry`
- `OrgDisableTwoFactorRequirementAuditEntry`
- `OrgEnableOAuthAppRestrictionsAuditEntry`
- `OrgEnableSamlAuditEntry`

- OrgEnableTwoFactorRequirementAuditEntry
- OrgInviteMemberAuditEntry
- OrgInviteToBusinessAuditEntry
- OrgOAuthAppAccessApprovedAuditEntry
- OrgOAuthAppAccessDeniedAuditEntry
- OrgOAuthAppAccessRequestedAuditEntry
- OrgRemoveBillingManagerAuditEntry
- OrgRemoveMemberAuditEntry
- OrgRemoveOutsideCollaboratorAuditEntry
- OrgRestoreMemberAuditEntry
- OrgRestoreMemberMembershipOrganizationAuditEntryData
- OrgUnblockUserAuditEntry
- OrgUpdateDefaultRepositoryPermissionAuditEntry
- OrgUpdateMemberAuditEntry
- OrgUpdateMemberRepositoryCreationPermissionAuditEntry
- OrgUpdateMemberRepositoryInvitationPermissionAuditEntry
- PrivateRepositoryForkingDisableAuditEntry
- PrivateRepositoryForkingEnableAuditEntry
- RepoAccessAuditEntry
- RepoAddMemberAuditEntry
- RepoAddTopicAuditEntry
- RepoArchivedAuditEntry
- RepoChangeMergeSettingAuditEntry
- RepoConfigDisableAnonymousGitAccessAuditEntry
- RepoConfigDisableCollaboratorsOnlyAuditEntry
- RepoConfigDisableContributorsOnlyAuditEntry
- RepoConfigDisableSockpuppetDisallowedAuditEntry
- RepoConfigEnableAnonymousGitAccessAuditEntry
- RepoConfigEnableCollaboratorsOnlyAuditEntry
- RepoConfigEnableContributorsOnlyAuditEntry
- RepoConfigEnableSockpuppetDisallowedAuditEntry
- RepoConfigLockAnonymousGitAccessAuditEntry
- RepoConfigUnlockAnonymousGitAccessAuditEntry
- RepoCreateAuditEntry
- RepoDestroyAuditEntry
- RepoRemoveMemberAuditEntry



- `RepoRemoveTopicAuditEntry`
- `RepositoryVisibilityChangeDisableAuditEntry`
- `RepositoryVisibilityChangeEnableAuditEntry`
- `TeamAddMemberAuditEntry`
- `TeamAddRepositoryAuditEntry`
- `TeamChangeParentTeamAuditEntry`
- `TeamRemoveMemberAuditEntry`
- `TeamRemoveRepositoryAuditEntry`

## Fields

Name	Description
<code>organization</code> ( <a href="#">Organization</a> )	The Organization associated with the Audit Entry.
<code>organizationName</code> ( <a href="#">String</a> )	The name of the Organization.
<code>organizationResourcePath</code> ( <a href="#">URI</a> )	The HTTP path for the organization.
<code>organizationUrl</code> ( <a href="#">URI</a> )	The HTTP URL for the organization.

## PackageOwner

Represents an owner of a package.

### Implemented by

- `Organization`
- `Repository`
- `User`

## Fields

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

Name	Description
packages ( <a href="#">Package Connection!</a> )	<p>A list of packages under the owner.</p> <div> <p><b>Arguments</b></p> <hr/> <p>after (<a href="#">String</a>) Returns the elements in the list that come after the specified cursor.</p> <hr/> <p>before (<a href="#">String</a>) Returns the elements in the list that come before the specified cursor.</p> <hr/> <p>first (<a href="#">Int</a>) Returns the first <math>n</math> elements from the list.</p> <hr/> <p>last (<a href="#">Int</a>) Returns the last <math>n</math> elements from the list.</p> <hr/> <p>names (<a href="#">[String]</a>) Find packages by their names.</p> <hr/> <p>orderBy (<a href="#">PackageOrder</a>) Ordering of the returned packages.</p> <hr/> <p>packageType (<a href="#">PackageType</a>) Filter registry package by type.</p> <hr/> <p>repositoryId (<a href="#">ID</a>) Find packages in a repository by ID.</p> </div>

## ProfileOwner

Represents any entity on GitHub that has a profile page.

### Implemented by

- [Organization](#)
- [User](#)

## Fields

Name	Description
<code>anyPinnableItems(Boolean!)</code>	Determine if this repository owner has any items that can be pinned to their profile. <div><b>Arguments</b><div>type (<code>PinnableItemType</code>) Filter to only a particular kind of pinnable item.</div></div>
<code>email (String)</code>	The public profile email.
<code>itemShowcase(ProfileItem Showcase!)</code>	Showcases a selection of repositories and gists that the profile owner has either curated or that have been selected automatically based on popularity.
<code>location (String)</code>	The public profile location.
<code>login (String!)</code>	The username used to login.
<code>name (String)</code>	The public profile name.

Name	Description
<code>pinnableItems(PinnableItemConnection!)</code>	<p>A list of repositories and gists this profile owner can pin to their profile.</p> <div><h3>Arguments</h3><p><code>after</code> (<code>String</code>) Returns the elements in the list that come after the specified cursor.</p><p><code>before</code> (<code>String</code>) Returns the elements in the list that come before the specified cursor.</p><p><code>first</code> (<code>Int</code>) Returns the first <math>n</math> elements from the list.</p><p><code>last</code> (<code>Int</code>) Returns the last <math>n</math> elements from the list.</p><p><code>types</code> (<code>[PinnableItemType!]</code>) Filter the types of pinnable items that are returned.</p></div>

Name	Description
<code>pinnedItems(PinnableItemConnection!)</code>	A list of repositories and gists this profile owner has pinned to their profile.
<div> <div>Arguments</div> <div> <div> after (<code>String</code>)  Returns the elements in the list that come after the specified cursor. </div> <div> before (<code>String</code>)  Returns the elements in the list that come before the specified cursor. </div> <div> first (<code>Int</code>)  Returns the first <math>n</math> elements from the list. </div> <div> last (<code>Int</code>)  Returns the last <math>n</math> elements from the list. </div> <div> types (<code>[PinnableItemType!]</code>)  Filter the types of pinned items that are returned. </div> </div> </div>	
<code>pinnedItemsRemaining(Int!)</code>	Returns how many more items this profile owner can pin to their profile.
<code>viewerCanChangePinnedItems(Boolean!)</code>	Can the viewer pin repositories and gists to the profile?.
<code>websiteUrl(Uri)</code>	The public profile website URL.

## ProjectOwner

Represents an owner of a Project.

### Implemented by

- `Organization`

- `Repository`
- `User`

Fields

Name	Description
project ( <code>Project</code> )	Find project by number. <div><div>Arguments</div><div>number (<code>Int!</code>) The project number to find.</div></div>

Name	Description
projects ( <a href="#">Project Connection!</a> )	<p>A list of projects under the owner.</p> <div> <p><b>Arguments</b></p> <hr/> <p>after (<a href="#">String</a>) Returns the elements in the list that come after the specified cursor.</p> <hr/> <p>before (<a href="#">String</a>) Returns the elements in the list that come before the specified cursor.</p> <hr/> <p>first (<a href="#">Int</a>) Returns the first <math>n</math> elements from the list.</p> <hr/> <p>last (<a href="#">Int</a>) Returns the last <math>n</math> elements from the list.</p> <hr/> <p>orderBy (<a href="#">ProjectOrder</a>) Ordering options for projects returned from the connection.</p> <hr/> <p>search (<a href="#">String</a>) Query to search projects by, currently only searching by name.</p> <hr/> <p>states (<a href="#">[ProjectState!]</a>) A list of states to filter the projects by.</p> </div>
projectsResourcePath ( <a href="#">URI!</a> )	The HTTP path listing owners projects.
projectsUrl ( <a href="#">URI!</a> )	The HTTP URL listing owners projects.
viewerCanCreateProjects( <a href="#">Boolean!</a> )	Can the current viewer create new projects on this owner.

## Reactable

Represents a subject that can be reacted on.

## Implemented by

- `CommitComment`
- `Issue`
- `IssueComment`
- `PullRequest`
- `PullRequestReview`
- `PullRequestReviewComment`
- `TeamDiscussion`
- `TeamDiscussionComment`

## Fields

Name	Description
<code>databaseId</code> ( <code>Int</code> )	Identifies the primary key from the database.
<code>reactionGroups</code> ( <code>[ReactionGroup!]</code> )	A list of reactions grouped by content left on the subject.



Name	Description
reactions ( <a href="#">ReactionConnection!</a> )	A list of Reactions left on the Issue.
<div> <div>Arguments</div> <div> <div>after (<a href="#">String</a>) Returns the elements in the list that come after the specified cursor.</div> <div>before (<a href="#">String</a>) Returns the elements in the list that come before the specified cursor.</div> <div>content (<a href="#">ReactionContent</a>) Allows filtering Reactions by emoji.</div> <div>first (<a href="#">Int</a>) Returns the first <math>n</math> elements from the list.</div> <div>last (<a href="#">Int</a>) Returns the last <math>n</math> elements from the list.</div> <div>orderBy (<a href="#">ReactionOrder</a>) Allows specifying the order in which reactions are returned.</div> </div> </div>	
viewerCanReact ( <a href="#">Boolean!</a> )	Can user react to this subject.

## RepositoryAuditEntryData

Metadata for an audit entry with action repo.\*.

### Implemented by

- [OrgRestoreMemberMembershipRepositoryAuditEntryData](#)
- [PrivateRepositoryForkingDisableAuditEntry](#)
- [PrivateRepositoryForkingEnableAuditEntry](#)
- [RepoAccessAuditEntry](#)

- `RepoAddMemberAuditEntry`
- `RepoAddTopicAuditEntry`
- `RepoArchivedAuditEntry`
- `RepoChangeMergeSettingAuditEntry`
- `RepoConfigDisableAnonymousGitAccessAuditEntry`
- `RepoConfigDisableCollaboratorsOnlyAuditEntry`
- `RepoConfigDisableContributorsOnlyAuditEntry`
- `RepoConfigDisableSockpuppetDisallowedAuditEntry`
- `RepoConfigEnableAnonymousGitAccessAuditEntry`
- `RepoConfigEnableCollaboratorsOnlyAuditEntry`
- `RepoConfigEnableContributorsOnlyAuditEntry`
- `RepoConfigEnableSockpuppetDisallowedAuditEntry`
- `RepoConfigLockAnonymousGitAccessAuditEntry`
- `RepoConfigUnlockAnonymousGitAccessAuditEntry`
- `RepoCreateAuditEntry`
- `RepoDestroyAuditEntry`
- `RepoRemoveMemberAuditEntry`
- `RepoRemoveTopicAuditEntry`
- `TeamAddRepositoryAuditEntry`
- `TeamRemoveRepositoryAuditEntry`

## Fields

Name	Description
<code>repository</code> ( <a href="#">Repository</a> )	The repository associated with the action.
<code>repositoryName</code> ( <a href="#">String</a> )	The name of the repository.
<code>repositoryResourcePath</code> ( <a href="#">URI</a> )	The HTTP path for the repository.
<code>repositoryUrl</code> ( <a href="#">URI</a> )	The HTTP URL for the repository.

## RepositoryInfo

A subset of repository info.

## Implemented by

- `Repository`

## Fields

Name	Description
<code>createdAt(DateTime!)</code>	Identifies the date and time when the object was created.
<code>description(String)</code>	The description of the repository.
<code>descriptionHTML(HTML!)</code>	The description of the repository rendered to HTML.
<code>forkCount(Int!)</code>	Returns how many forks there are of this repository in the whole network.
<code>hasIssuesEnabled(Boolean!)</code>	Indicates if the repository has issues feature enabled.
<code>hasProjectsEnabled(Boolean!)</code>	Indicates if the repository has the Projects feature enabled.
<code>hasWikiEnabled(Boolean!)</code>	Indicates if the repository has wiki feature enabled.
<code>homepageUrl(URI)</code>	The repository's URL.
<code>isArchived(Boolean!)</code>	Indicates if the repository is unmaintained.
<code>isFork(Boolean!)</code>	Identifies if the repository is a fork.
<code>isInOrganization(Boolean!)</code>	Indicates if a repository is either owned by an organization, or is a private fork of an organization repository.
<code>isLocked(Boolean!)</code>	Indicates if the repository has been locked or not.
<code>isMirror(Boolean!)</code>	Identifies if the repository is a mirror.
<code>isPrivate(Boolean!)</code>	Identifies if the repository is private.

Name	Description
isTemplate( <a href="#">Boolean!</a> )	Identifies if the repository is a template that can be used to generate new repositories.
licenseInfo( <a href="#">License</a> )	The license associated with the repository.
lockReason( <a href="#">RepositoryLock Reason</a> )	The reason the repository has been locked.
mirrorUrl ( <a href="#">URI</a> )	The repository's original mirror URL.
name ( <a href="#">String!</a> )	The name of the repository.
nameWithOwner( <a href="#">String!</a> )	The repository's name with owner.
openGraphImageUrl( <a href="#">URI!</a> )	The image used to represent this repository in Open Graph data.
owner( <a href="#">RepositoryOwner!</a> )	The User owner of the repository.
pushedAt ( <a href="#">DateTime</a> )	Identifies when the repository was last pushed to.
resourcePath ( <a href="#">URI!</a> )	The HTTP path for this repository.
shortDescription HTML ( <a href="#">HTML!</a> )	A description of the repository, rendered to HTML without any links in it.
<div> <div>Arguments</div> <div> <div>limit (<a href="#">Int</a>)</div> <div>How many characters to return.</div> <div>The default value is 200.</div> </div> </div>	
updatedAt( <a href="#">DateTime!</a> )	Identifies the date and time when the object was last updated.
url ( <a href="#">URI!</a> )	The HTTP URL for this repository.

Name	Description
usesCustomOpenGraphImage( <a href="#">Boolean!</a> )	Whether this repository has a custom image to use with Open Graph as opposed to being represented by the owner's avatar.

## RepositoryNode

Represents an object that belongs to a repository.

### Implemented by

- [CommitComment](#)
- [CommitCommentThread](#)
- [Issue](#)
- [IssueComment](#)
- [PullRequest](#)
- [PullRequestCommitCommentThread](#)
- [PullRequestReview](#)
- [PullRequestReviewComment](#)
- [RepositoryVulnerabilityAlert](#)

### Fields

Name	Description
repository ( <a href="#">Repository!</a> )	The repository associated with this node.

## RepositoryOwner

Represents an owner of a Repository.

### Implemented by

- `Organization`
- `User`

## Fields

Name	Description
<code>avatarUrl(<a href="#">URI!</a>)</code>	A URL pointing to the owner's public avatar. <div><b>Arguments</b><div><code>size (<a href="#">Int</a>)</code> The size of the resulting square image.</div></div>
<code>login (<a href="#">String!</a>)</code>	The username used to login.

Name	Description
repositories( <a href="#">Repository Connection!</a> )	<p>A list of repositories that the user owns.</p> <div> <h3>Arguments</h3> <div> <p><b>affiliations</b> (<a href="#">[RepositoryAffiliation]</a>)  Array of viewer's affiliation options for repositories returned from the connection. For example, OWNER will include only repositories that the current viewer owns.</p> <p><b>after</b> (<a href="#">String</a>)  Returns the elements in the list that come after the specified cursor.</p> <p><b>before</b> (<a href="#">String</a>)  Returns the elements in the list that come before the specified cursor.</p> <p><b>first</b> (<a href="#">Int</a>)  Returns the first <math>n</math> elements from the list.</p> <p><b>isFork</b> (<a href="#">Boolean</a>)  If non-null, filters repositories according to whether they are forks of another repository.</p> <p><b>isLocked</b> (<a href="#">Boolean</a>)  If non-null, filters repositories according to whether they have been locked.</p> <p><b>last</b> (<a href="#">Int</a>)  Returns the last <math>n</math> elements from the list.</p> <p><b>orderBy</b> (<a href="#">RepositoryOrder</a>)  Ordering options for repositories returned from the connection.</p> <p><b>ownerAffiliations</b> (<a href="#">[RepositoryAffiliation]</a>)  Array of owner's affiliation options for repositories returned from the connection. For example, OWNER will include only repositories that the organization or user being viewed owns.</p> <p><b>privacy</b> (<a href="#">RepositoryPrivacy</a>)  If non-null, filters repositories according to privacy.</p> </div> </div>

Name	Description
repository( <a href="#">Repository</a> )	Find Repository.
<div> <div>Arguments</div> <div> name (<a href="#">String!</a>)  Name of Repository to find. </div> </div>	
resourcePath( <a href="#">URI!</a> )	The HTTP URL for the owner.
url ( <a href="#">URI!</a> )	The HTTP URL for the owner.

## Sponsorable

Entities that can be sponsored through GitHub Sponsors.

### Implemented by

- [Organization](#)
- [User](#)

### Fields

Name	Description
sponsorsListing( <a href="#">SponsorsListing</a> )	The GitHub Sponsors listing for this user.



Name	Description
sponsorshipsAsMaintainer( <a href="#">SponsorshipConnection!</a> )	<p>This object's sponsorships as the maintainer.</p> <div> <p><b>Arguments</b></p> <hr/> <p><b>after</b> (<a href="#">String</a>) Returns the elements in the list that come after the specified cursor.</p> <hr/> <p><b>before</b> (<a href="#">String</a>) Returns the elements in the list that come before the specified cursor.</p> <hr/> <p><b>first</b> (<a href="#">Int</a>) Returns the first <math>n</math> elements from the list.</p> <hr/> <p><b>includePrivate</b> (<a href="#">Boolean</a>) Whether or not to include private sponsorships in the result set. The default value is false.</p> <hr/> <p><b>last</b> (<a href="#">Int</a>) Returns the last <math>n</math> elements from the list.</p> <hr/> <p><b>orderBy</b> (<a href="#">SponsorshipOrder</a>) Ordering options for sponsorships returned from this connection. If left blank, the sponsorships will be ordered based on relevancy to the viewer.</p> </div>

Name	Description
sponsorshipsAs Sponsor( <a href="#">SponsorshipConnection!</a> )	This object's sponsorships as the sponsor. <div> <h3>Arguments</h3> <div> <p>after (<a href="#">String</a>)</p> <p>Returns the elements in the list that come after the specified cursor.</p> </div> <div> <p>before (<a href="#">String</a>)</p> <p>Returns the elements in the list that come before the specified cursor.</p> </div> <div> <p>first (<a href="#">Int</a>)</p> <p>Returns the first <math>n</math> elements from the list.</p> </div> <div> <p>last (<a href="#">Int</a>)</p> <p>Returns the last <math>n</math> elements from the list.</p> </div> <div> <p>orderBy (<a href="#">SponsorshipOrder</a>)</p> <p>Ordering options for sponsorships returned from this connection. If left blank, the sponsorships will be ordered based on relevancy to the viewer.</p> </div> </div>

## Starrable

Things that can be starred.

### Implemented by

- [Gist](#)
- [Repository](#)
- [Topic](#)

### Fields

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

Name	Description
stargazerCount ( <a href="#">Int!</a> )	Returns a count of how many stargazers there are on this object.
stargazers ( <a href="#">Stargazer Connection!</a> )	<p>A list of users who have starred this starrable.</p> <div> <p><b>Arguments</b></p> <hr/> <p>after (<a href="#">String</a>) Returns the elements in the list that come after the specified cursor.</p> <hr/> <p>before (<a href="#">String</a>) Returns the elements in the list that come before the specified cursor.</p> <hr/> <p>first (<a href="#">Int</a>) Returns the first <math>n</math> elements from the list.</p> <hr/> <p>last (<a href="#">Int</a>) Returns the last <math>n</math> elements from the list.</p> <hr/> <p>orderBy (<a href="#">StarOrder</a>) Order for connection.</p> </div>
viewerHasStarred( <a href="#">Boolean!</a> )	Returns a boolean indicating whether the viewing user has starred this starrable.

## Subscribable

Entities that can be subscribed to for web and email notifications.

### Implemented by

- [Commit](#)
- [Issue](#)
- [PullRequest](#)
- [Repository](#)

- `Team`
- `TeamDiscussion`

## Fields

Name	Description
<code>viewerCanSubscribe(Boolean!)</code>	Check if the viewer is able to change their subscription status for the repository.
<code>viewerSubscription(SubscriptionState)</code>	Identifies if the viewer is watching, not watching, or ignoring the subscribable entity.

## TeamAuditEntryData

Metadata for an audit entry with action team.\*.

## Implemented by

- `OrgRestoreMemberMembershipTeamAuditEntryData`
- `TeamAddMemberAuditEntry`
- `TeamAddRepositoryAuditEntry`
- `TeamChangeParentTeamAuditEntry`
- `TeamRemoveMemberAuditEntry`
- `TeamRemoveRepositoryAuditEntry`

## Fields

Name	Description
<code>team(Team)</code>	The team associated with the action.
<code>teamName(String)</code>	The name of the team.
<code>teamResourcePath(Uri)</code>	The HTTP path for this team.

Name	Description
teamUrl ( <a href="#">URI</a> )	The HTTP URL for this team.

## TopicAuditEntryData

Metadata for an audit entry with a topic.

### Implemented by

- [RepoAddTopicAuditEntry](#)
- [RepoRemoveTopicAuditEntry](#)

### Fields

Name	Description
topic ( <a href="#">Topic</a> )	The name of the topic added to the repository.
topicName ( <a href="#">String</a> )	The name of the topic added to the repository.

## UniformResourceLocatable

Represents a type that can be retrieved by a URL.

### Implemented by

- [Bot](#)
- [CheckRun](#)
- [ClosedEvent](#)
- [Commit](#)
- [ConvertToDraftEvent](#)
- [CrossReferencedEvent](#)

- `Gist`
- `Issue`
- `Mannequin`
- `MergedEvent`
- `Milestone`
- `Organization`
- `PullRequest`
- `PullRequestCommit`
- `ReadyForReviewEvent`
- `Release`
- `Repository`
- `RepositoryTopic`
- `ReviewDismissedEvent`
- `TeamDiscussion`
- `TeamDiscussionComment`
- `User`

## Fields

Name	Description
<code>resourcePath (URI!)</code>	The HTML path to this resource.
<code>url (URI!)</code>	The URL to this resource.

## Updatable

Entities that can be updated.

## Implemented by

- `CommitComment`
- `GistComment`
- `Issue`
- `IssueComment`

- `Project`
- `PullRequest`
- `PullRequestReview`
- `PullRequestReviewComment`
- `TeamDiscussion`
- `TeamDiscussionComment`

## Fields

Name	Description
<code>viewerCanUpdate</code> ( <code>Boolean!</code> )	Check if the current viewer can update this object.

## UpdatableComment

Comments that can be updated.

## Implemented by

- `CommitComment`
- `GistComment`
- `Issue`
- `IssueComment`
- `PullRequest`
- `PullRequestReview`
- `PullRequestReviewComment`
- `TeamDiscussion`
- `TeamDiscussionComment`

## Fields

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

Name	Description
viewerCannotUpdateReasons ([ <a href="#">CommentCannotUpdateReason!</a> ]!)	Reasons why the current viewer can not update this comment.