API Endpoints Documentation

Repository: CodeIgniter4

File: DCO.txt

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(1) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

(2) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or

(3) The contribution was provided directly to me by some other person who certified (1), (2) or (3) and I have not modified it.

(4) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

Repository: content

File: content-bug.md

```
---
name: Content bug
about: "Use this template to notify us of a content bug that needs fixing"
labels: "needs-triage"
assignees:
title: "Content bug: <TITLE OF PROBLEM>"
---

## What page(s) did you find the problem on?

<!-- include the URL or URLs where you found the problem. If it is a widespread
problem over many pages, just give us a couple of example URLs rather than the
whole lot  -->

## Specific page section or heading?

<!-- include the specific heading underneath which the problem can be found, if
relevant, to help us locate the problem more easily  -->

## What is the problem?

<!-- include a description of the problem ? is some text misspelt, or
inaccurate? Does an example not work? Is the document missing some information?
Is something just weird?  -->
```

```
## What did you expect to see?

<!-- If you have an idea of what the solution to your problem is, please
provide details here. If you don't know, then that's OK   -->

## Did you test this? If so, how?

<!-- Please provide any steps you took to test the problem, if appropriate  -->
File: ff-project-issue.md
---
name: Firefox Release Issue
about: "Use this template to create an issue for a Firefox Release Project"
labels: ""
assignees:
title: ""
---

## Acceptance Criteria

- [ ] The listed features are documented sufficiently on MDN
- [ ] BCD is updated
- [ ] Interactive example and data repos are updated if appropriate
- [ ] The content has been reviewed as needed

### For folks helping with Firefox related documentation

- [ ] Set bugs to `dev-doc-complete`
```

- [ ] Add entry to Firefox release notes if feature is enabled in release

or

- [ ] Add entry to Firefox experimental features page if feature is not yet enabled in release

Features to document

Related Gecko bugs

Other

File: new-content-suggestion.md

---

name: New content suggestion

about: "Use this template to suggest new larger areas of work that you think the MDN writers should work on"

labels: "Opportunity assessment"

assignees: chrisdavidmills

title: "Content suggestion: <TITLE OF SUGGESTION>"

What is the new suggestion?

<!-- include a short description of the content work suggestion -->

Why is it important or useful?

<!-- Tell us why the idea is important or useful. Include any information you can think of that would be useful, for example:

How many pages are likely to be needed?

How much time do you think this work should take? A few hours? A week? A month?

Will the work enable learners or professionals to achieve their goals better?

Does it address critical needs in the web industry?

Is the work an operational necessity, i.e. is not having it a security risk?

Does the content help make the web more ethical?

 -->

Other supporting information

<!-- include any other useful supporting information, such as spec URL, explainer URL, link to further supporting information -->

Repository: fig-standards

File: CONTRIBUTING.md

# Contributing to the PHP-FIG

Anybody who subscribes to the Google Group, is part of the PHP-FIG. As soon as you subscribe to the [mailing list](http://groups.google.com/group/php-fig/) and/or join the [IRC channel](http://www.php-fig.org/irc/) you are a PHP-FIG Community Member, who can influence standards, make suggestions, give feedback, etc. Only PHP-FIG Voting Members can start or participate in votes, but the discussion and formation stages involve everyone.

See the [PHP-FIG FAQ](http://www.php-fig.org/faqs/) for more information.

# Licensing

By contributing code you agree to license your contribution under the MIT license.

By contributing documentation, examples, or any other non-code assets you agree to license your contribution under the CC BY 3.0 license. Attribution shall be given according to the current bylaws of this group.

# Merge & Access Policy

All Editors, Coordinators and Sponsors of specifications in draft & review stage have access to push to this (php-fig/fig-standards) repository; subject to secretary discretion

All Editors, Coordinators and Sponsors of specifications have push access to utility and interface repositories and retrain this even after acceptance; subject to secretary discretion.

The master branch of all repositories are protected and therefore cannot be forced pushed to.

Secretaries have, and are the only ones to have, full administrative access to all repositories and to the github organisation.

Guidelines for merging

* Never force push to any branch on a repository in the php-fig organisation
* All changes must go through pull requests, a commit should never be pushed directly (excluding inital commits on new interface/util repositories) to master
* All pull requests relating to a draft PSR must be approved (with a formal +1 comment) or merged by the PSR Editor, except in the review phase when the coordinator should seek commment from the editor, but merging is at the coordinator's discretion
* You must never merge a pull request that affects any file in the repository other than those you are on a working group for; you should request a secretary or member of that working group (mention @php-fig/psr-x) do so
* You should never merge your own pull request
* A change should never be merged to an accepted PSR without approval from secretaries, who will attempt to seek confirmation from the former Editors
* A change to bylaws shouldn't be merged by anyone other than a secretary
* Pull requests may be triaged (have labels applied) by anyone with push access,

no matter which PSR they are on the working group for or which PSR it affects; but they cannot close a pull request or issue affecting other PSRs
* After approval of a specification, all merges to an interface or utility repository must be approved by a secretary; who is required to give suitable notice and seek comment from the working group team, but it is not required that they approve
* Tags on utility and interface repositories should be created and PGP signed by Secretaries, who should publish their PGP public keys and register them on github

These guidelines are subject to exceptions and changes at secretaries discretion. Should you have any questions please contact the secretaries on info [at] php-fig [dot] org. Failure to comply with guidelines will result in revokation of merge access. Merge access is a privilege and not a right.

# Tagging

Tagging on utility and interface repository should be done regularly, ideally after every merge, or every batch of merges after PSR approval.

Versioning should follow semantic versioning and primarily just be simple patch fix increments (following semantic versioning). The first 1.0.0 tag should be created on PSR approval.

All tags should be PGP signed.

A changelog should be provided including a list of changes, crediting the contributor and referencing the pull request/issue.

File: LICENSE-CC.md

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

    i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
    ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the

exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly

Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and

enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at http://creativecommons.org/.

File: LICENSE-MIT.md

Copyright (c) 2013-2017 PHP Framework Interop Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

File: LICENSE.md

# License

Repository: FOSCommentBundle

File: AclCommentManager.php

```
<?php

/*
 * This file is part of the FOSCommentBundle package.
 *
 * (c) FriendsOfSymfony <http://friendsofsymfony.github.com/>
 *
 * This source file is subject to the MIT license that is bundled
 * with this source code in the file LICENSE.
 */

namespace FOS\CommentBundle\Acl;

use FOS\CommentBundle\Model\CommentInterface;
use FOS\CommentBundle\Model\CommentManagerInterface;
use FOS\CommentBundle\Model\ThreadInterface;
use Symfony\Component\Security\Core\Exception\AccessDeniedException;

/**
 * Wraps a real implementation of CommentManagerInterface and
 * performs Acl checks with the configured Comment Acl service.
 *
 * @author Tim Nagel <tim@nagel.com.au
 */
class AclCommentManager implements CommentManagerInterface
```

```
{
    /**
     * The CommentManager instance to be wrapped with ACL.
     *
     * @var CommentManagerInterface
     */
    protected $realManager;

    /**
     * The CommentAcl instance for checking permissions.
     *
     * @var CommentAclInterface
     */
    protected $commentAcl;

    /**
     * The ThreadAcl instance for checking permissions.
     *
     * @var ThreadAclInterface
     */
    protected $threadAcl;

    /**
     * Constructor.
     *
        * @param CommentManagerInterface $commentManager The concrete CommentManager
service
```

```
     * @param CommentAclInterface     $commentAcl     The Comment Acl service
     * @param ThreadAclInterface      $threadAcl      The Thread Acl service
     */
    public function __construct(CommentManagerInterface $commentManager,
CommentAclInterface $commentAcl, ThreadAclInterface $threadAcl)
    {
        $this->realManager = $commentManager;
        $this->commentAcl = $commentAcl;
        $this->threadAcl = $threadAcl;
    }

    /**
     * {@inheritdoc}
     *
     * @throws AccessDeniedException
     */
    public function findCommentTreeByThread(ThreadInterface $thread, $sorter = null,
$depth = null)
    {
        $comments = $this->realManager->findCommentTreeByThread($thread, $sorter,
$depth);

        if (!$this->authorizeViewCommentTree($comments)) {
            throw new AccessDeniedException();
        }

        return $comments;
    }

    /**
     * {@inheritdoc}
     */
    public function findCommentsByThread(ThreadInterface $thread, $depth = null,
$sorterAlias = null)
    {
        $comments = $this->realManager->findCommentsByThread($thread, $depth,
$sorterAlias);

        foreach ($comments as $comment) {
            if (!$this->commentAcl->canView($comment)) {
                throw new AccessDeniedException();
            }
        }

        return $comments;
    }

    /**
     * {@inheritdoc}
     */
    public function findCommentTreeByCommentId($commentId, $sorter = null)
    {
        $comments = $this->realManager->findCommentTreeByCommentId($commentId, $sorter);
        if (!$this->authorizeViewCommentTree($comments)) {
            throw new AccessDeniedException();
        }

        return $comments;
    }

    /**
     * {@inheritdoc}
     */
    public function saveComment(CommentInterface $comment)
    {
        if (!$this->threadAcl->canView($comment->getThread())) {
            throw new AccessDeniedException();
        }

        if (!$this->commentAcl->canReply($comment->getParent())) {
            throw new AccessDeniedException();
        }

        $newComment = $this->isNewComment($comment);

        if (!$newComment && !$this->commentAcl->canEdit($comment)) {
            throw new AccessDeniedException();
        }

         if (($comment::STATE_DELETED === $comment->getState() || $comment::STATE_DELETED
=== $comment->getPreviousState())
            && !$this->commentAcl->canDelete($comment)
        ) {
            throw new AccessDeniedException();
        }

        $state = $this->realManager->saveComment($comment);

        if ($state && $newComment) {
            $this->commentAcl->setDefaultAcl($comment);
        }

        return $state;
    }

    /**
     * {@inheritdoc}
     **/
    public function findCommentById($id)
    {
        $comment = $this->realManager->findCommentById($id);

        if (null !== $comment && !$this->commentAcl->canView($comment)) {
            throw new AccessDeniedException();
        }

        return $comment;
    }

    /**
     * {@inheritdoc}
     */
    public function createComment(ThreadInterface $thread, CommentInterface $parent =
null)
    {
        return $this->realManager->createComment($thread, $parent);
    }

    /**
     * {@inheritdoc}
     */
    public function isNewComment(CommentInterface $comment)
    {
        return $this->realManager->isNewComment($comment);
    }

    /**
     * {@inheritdoc}
     */
    public function getClass()
    {
        return $this->realManager->getClass();
    }
    /**
     * Iterates over a comment tree array and makes sure all comments
     * have appropriate view permissions.
     *
     * @param array $comments A comment tree
     *
     * @return bool
     */
    protected function authorizeViewCommentTree(array $comments)
    {
        foreach ($comments as $comment) {
            if (!$this->commentAcl->canView($comment['comment'])) {
                return false;
            }

            if (is_array($comment['children'])) {
                return $this->authorizeViewCommentTree($comment['children']);
            }
        }

        return true;
    }
}
```

File: AclThreadManager.php

```
<?php

/*
```

```
 * This file is part of the FOSCommentBundle package.
 *
 * (c) FriendsOfSymfony <http://friendsofsymfony.github.com/>
 *
 * This source file is subject to the MIT license that is bundled
 * with this source code in the file LICENSE.
 */

namespace FOS\CommentBundle\Acl;

use FOS\CommentBundle\Model\ThreadInterface;
use FOS\CommentBundle\Model\ThreadManagerInterface;
use Symfony\Component\Security\Core\Exception\AccessDeniedException;

/**
 * Wraps a real implementation of ThreadManagerInterface and
 * performs Acl checks with the configured Thread Acl service.
 *
 * @author Tim Nagel <tim@nagel.com.au
 */
class AclThreadManager implements ThreadManagerInterface
{
    /**
     * The ThreadManager instance to be wrapped with ACL.
     *
     * @var ThreadManagerInterface
     */
    protected $realManager;


    /**
     * The Thread Acl instance for querying Acls.
     *
     * @var ThreadAclInterface
     */
    protected $threadAcl;


    /**
     * Constructor.
     *
     * @param ThreadManagerInterface $threadManager The concrete ThreadManager service
     * @param ThreadAclInterface     $threadAcl     The Thread Acl service
     */
           public    function    __construct(ThreadManagerInterface    $threadManager,
ThreadAclInterface $threadAcl)
    {
        $this->realManager = $threadManager;
        $this->threadAcl = $threadAcl;
    }


    /**
     * {@inheritdoc}
     */
    public function findThreadById($id)
    {
```

```
    $thread = $this->realManager->findThreadById($id);

    if (null !== $thread && !$this->threadAcl->canView($thread)) {
        throw new AccessDeniedException();
    }

    return $thread;
}

/**
 * {@inheritdoc}
 */
public function findThreadBy(array $criteria)
{
    $thread = $this->realManager->findThreadBy($criteria);

    if (null !== $thread && !$this->threadAcl->canView($thread)) {
        throw new AccessDeniedException();
    }

    return $thread;
}

/**
 * {@inheritdoc}
 */
public function findThreadsBy(array $criteria)
```

```
{
    $threads = $this->realManager->findThreadsBy($criteria);

    foreach ($threads as $thread) {
        if (!$this->threadAcl->canView($thread)) {
            throw new AccessDeniedException();
        }
    }

    return $threads;
}

/**
 * {@inheritdoc}
 */
public function findAllThreads()
{
    $threads = $this->realManager->findAllThreads();

    foreach ($threads as $thread) {
        if (!$this->threadAcl->canView($thread)) {
            throw new AccessDeniedException();
        }
    }

    return $threads;
}
```

```
/**
 * {@inheritdoc}
 */
public function createThread($id = null)
{
    return $this->realManager->createThread($id);
}

/**
 * {@inheritdoc}
 */
public function saveThread(ThreadInterface $thread)
{
    if (!$this->threadAcl->canCreate()) {
        throw new AccessDeniedException();
    }

    $newThread = $this->isNewThread($thread);

    if (!$newThread && !$this->threadAcl->canEdit($thread)) {
        throw new AccessDeniedException();
    }

    $this->realManager->saveThread($thread);

    if ($newThread) {
```

```php
            $this->threadAcl->setDefaultAcl($thread);
        }
    }

    /**
     * {@inheritdoc}
     */
    public function isNewThread(ThreadInterface $thread)
    {
        return $this->realManager->isNewThread($thread);
    }

    /**
     * {@inheritdoc}
     */
    public function getClass()
    {
        return $this->realManager->getClass();
    }
}
```

File: AclVoteManager.php

```php
<?php

/*
 * This file is part of the FOSCommentBundle package.
 *
 * (c) FriendsOfSymfony <http://friendsofsymfony.github.com/>
 *
 * This source file is subject to the MIT license that is bundled
 * with this source code in the file LICENSE.
 */

namespace FOS\CommentBundle\Acl;

use FOS\CommentBundle\Model\VotableCommentInterface;
use FOS\CommentBundle\Model\VoteInterface;
use FOS\CommentBundle\Model\VoteManagerInterface;
use Symfony\Component\Security\Core\Exception\AccessDeniedException;

/**
 * Wraps a real implementation of VoteManagerInterface and
 * performs Acl checks with the configured Vote Acl service.
 *
 * @author Tim Nagel <tim@nagel.com.au
 */
class AclVoteManager implements VoteManagerInterface
{
    /**
     * The VoteManager instance to be wrapped with ACL.
     *
     * @var VoteManagerInterface
     */
    protected $realManager;
    /**
     * The Vote Acl instance for querying Acls.
     *
     * @var VoteAclInterface
     */
    protected $voteAcl;

    /**
     * The Comment Acl for querying Acls.
     *
     * @var CommentAclInterface
     */
    protected $commentAcl;

    /**
     * {@inheritdoc}
     */
       public  function  __construct(VoteManagerInterface  $voteManager,  VoteAclInterface
$voteAcl, CommentAclInterface $commentAcl)
    {
        $this->realManager = $voteManager;
        $this->voteAcl = $voteAcl;
        $this->commentAcl = $commentAcl;
    }

    /**
     * {@inheritdoc}
```

```
 */
public function findVoteById($id)
{
    $vote = $this->realManager->findVoteById($id);

    if ($vote && !$this->voteAcl->canView($vote)) {
        throw new AccessDeniedException();
    }

    return $vote;
}

/**
 * {@inheritdoc}
 */
public function findVoteBy(array $criteria)
{
    $vote = $this->realManager->findVoteBy($criteria);

    if ($vote && !$this->voteAcl->canView($vote)) {
        throw new AccessDeniedException();
    }

    return $vote;
}

/**
```

```
 * {@inheritdoc}
 */
public function findVotesByComment(VotableCommentInterface $comment)
{
    $votes = $this->realManager->findVotesByComment($comment);

    foreach ($votes as $vote) {
        if (!$this->voteAcl->canView($vote)) {
            throw new AccessDeniedException();
        }
    }

    return $votes;
}

/**
 * {@inheritdoc}
 */
public function saveVote(VoteInterface $vote)
{
    if (!$this->voteAcl->canCreate()) {
        throw new AccessDeniedException();
    }

    if (!$this->commentAcl->canView($vote->getComment())) {
        throw new AccessDeniedException();
    }
```

```php
        $this->realManager->saveVote($vote);
        $this->voteAcl->setDefaultAcl($vote);
    }

    /**
     * {@inheritdoc}
     */
    public function createVote(VotableCommentInterface $comment)
    {
        return $this->realManager->createVote($comment);
    }

    /**
     * {@inheritdoc}
     */
    public function getClass()
    {
        return $this->realManager->getClass();
    }
}
```

File: CommentAclInterface.php

```php
<?php

/*
 * This file is part of the FOSCommentBundle package.
 *
 * (c) FriendsOfSymfony <http://friendsofsymfony.github.com/>
 *
 * This source file is subject to the MIT license that is bundled
 * with this source code in the file LICENSE.
 */

namespace FOS\CommentBundle\Acl;

use FOS\CommentBundle\Model\CommentInterface;

/**
 * Used for checking if the ACL system will allow specific actions
 * to occur.
 *
 * @author Tim Nagel <tim@nagel.com.au>
 */
interface CommentAclInterface
{
    /**
     * Checks if the user should be able to create a comment.
     *
     * @return bool
     */
    public function canCreate();

    /**
     * Checks if the user should be able to view a comment.
     *
     * @param CommentInterface $comment
     *
     * @return bool
     */
    public function canView(CommentInterface $comment);

    /**
     * Checks if the user can reply to the supplied 'parent' comment
     * or if not supplied, just the ability to reply.
     *
     * @param CommentInterface|null $parent
     *
     * @return bool
     */
    public function canReply(CommentInterface $parent = null);

    /**
     * Checks if the user should be able to edit a comment.
     *
     * @param CommentInterface $comment
     *
     * @return bool
     */
    public function canEdit(CommentInterface $comment);

    /**
```

```php
 * Checks if the user should be able to delete a comment.
 *
 * @param CommentInterface $comment
 *
 * @return bool
 */
public function canDelete(CommentInterface $comment);

/**
 * Sets the default Acl permissions on a comment.
 *
 * Note: this does not remove any existing Acl and should only
 * be called on new CommentInterface instances.
 *
 * @param CommentInterface $comment
 *
 * @return void
 */
public function setDefaultAcl(CommentInterface $comment);

/**
 * Installs the Default 'fallback' Acl entries for generic access.
 *
 * @return void
 */
public function installFallbackAcl();
```

```php
    /**
     * Removes default Acl entries.
     *
     * @return void
     */
    public function uninstallFallbackAcl();
}
```

File: RoleCommentAcl.php

```php
<?php

/*
 * This file is part of the FOSCommentBundle package.
 *
 * (c) FriendsOfSymfony <http://friendsofsymfony.github.com/>
 *
 * This source file is subject to the MIT license that is bundled
 * with this source code in the file LICENSE.
 */

namespace FOS\CommentBundle\Acl;

use FOS\CommentBundle\Model\CommentInterface;
use Symfony\Component\Security\Core\Authorization\AuthorizationCheckerInterface;

/**
 * Implements Role checking using the Symfony Security component.
 *
 * @author Tim Nagel <tim@nagel.com.au>
 */
class RoleCommentAcl implements CommentAclInterface
{
    /**
     * @var AuthorizationCheckerInterface
     */
    private $authorizationChecker;

    /**
     * The FQCN of the Comment object.
     *
     * @var string
     */
    private $commentClass;

    /**
     * The role that will grant create permission for a comment.
     *
     * @var string
     */
    private $createRole;

    /**
     * The role that will grant view permission for a comment.
     *
     * @var string
     */
    private $viewRole;

    /**
     * The role that will grant edit permission for a comment.
     *
     * @var string
     */
    private $editRole;

    /**
     * The role that will grant delete permission for a comment.
     *
     * @var string
     */
    private $deleteRole;

    /**
     * Constructor.
     *
     * @param AuthorizationCheckerInterface $authorizationChecker
     * @param string                        $createRole
     * @param string                        $viewRole
     * @param string                        $editRole
     * @param string                        $deleteRole
     * @param string                        $commentClass
     */
```

```
public function __construct(AuthorizationCheckerInterface $authorizationChecker,
                            $createRole,
                            $viewRole,
                            $editRole,
                            $deleteRole,
                            $commentClass
) {
    $this->authorizationChecker = $authorizationChecker;
    $this->createRole = $createRole;
    $this->viewRole = $viewRole;
    $this->editRole = $editRole;
    $this->deleteRole = $deleteRole;
    $this->commentClass = $commentClass;
}

/**
 * Checks if the Security token has an appropriate role to create a new Comment.
 *
 * @return bool
 */
public function canCreate()
{
    return $this->authorizationChecker->isGranted($this->createRole);
}

/**
 * Checks if the Security token is allowed to view the specified Comment.
 *
 * @param CommentInterface $comment
 *
 * @return bool
 */
public function canView(CommentInterface $comment)
{
    return $this->authorizationChecker->isGranted($this->viewRole);
}

/**
 * Checks if the Security token is allowed to reply to a parent comment.
 *
 * @param CommentInterface|null $parent
 *
 * @return bool
 */
public function canReply(CommentInterface $parent = null)
{
    if (null !== $parent) {
        return $this->canCreate() && $this->canView($parent);
    }

    return $this->canCreate();
}

/**
```

```
     * Checks if the Security token has an appropriate role to edit the supplied Comment.
     *
     * @param CommentInterface $comment
     *
     * @return bool
     */
    public function canEdit(CommentInterface $comment)
    {
        return $this->authorizationChecker->isGranted($this->editRole);
    }

    /**
     * Checks if the Security token is allowed to delete a specific Comment.
     *
     * @param CommentInterface $comment
     *
     * @return bool
     */
    public function canDelete(CommentInterface $comment)
    {
        return $this->authorizationChecker->isGranted($this->deleteRole);
    }

    /**
     * Role based Acl does not require setup.
     *
     * @param CommentInterface $comment
     *
     * @return void
     */
    public function setDefaultAcl(CommentInterface $comment)
    {
    }

    /**
     * Role based Acl does not require setup.
     *
     * @return void
     */
    public function installFallbackAcl()
    {
    }

    /**
     * Role based Acl does not require setup.
     *
     * @return void
     */
    public function uninstallFallbackAcl()
    {
    }
}
```

File: Changelog.md

Changelog
=========

3.0.0 (2020-01-08)

* Dropped PHP 5, 7.1 support.
* Switched to short array syntax.
* Dropped Symfony 2 support.
* Dropped Symfony < 3.4 support.
* Dropped Symfony < 4.3 support.
* Removed `AbstractVote` fixture.
* Removed legacy form helpers.
* Add type hints and return types to several classes.
* Marked form factories final.
* Marked `CommentExtension` final.
* Removed deprecated `CommentExtension::isCommentDeleted` method.

2.4.0 (2020-01-08)

* Deprecated the `CommentExtension::isCommentDeleted` method.

2.3.3 (2019-11-06)

* Fix: use `Symfony\Contracts\EventDispatcher\Event` if available.
* Fix: use `LegacyEventDispatcherProxy` if available.

2.3.2 (2019-08-22)

* Fix: `AclCommentManager::saveComment` must have a return value.

2.3.1 (2019-07-05)

* Auto-injection of the container is deprecated since Symfony 4.2.

2.3.0 (2019-05-02)

* Removed deprecated transchoice tag.
* Updated deprecated routing syntax.
* Updated deprecated `TreeBuilder` usage.
* Updated deprecated unit test warnings.
* Switched all Twig classes to use PHP namespaces.
* If available, `ThreadController` will extend `AbstractController`.

2.2.1 (2019-03-29)

* Commands should use the manager alias.
* Fixed Romanian translation.
* Allow installation of `JMSSerializerBundle 3.x`.

2.2.0 (2018-06-04)

* Allow template override in Symfony 4.
* Commands are now lazy loaded.
* Service class parameters have been removed.

* Added autowire support for model managers.

2.1.0 (2018-03-04)

* Dropped PHP < 5.6 support.

* Dropped Symfony < 2.7 support.

* Dropped jQuery < 3 support.

* Dropped HHVM support.

* Added Symfony 4 support.

* Signature of `ThreadPermalinkListener` has been changed.

* Route `ThreadController::getThreadsActions` throws 404 if called without id's.

* Added pipeline parser to integrate several parsers together.

* Added flat view for async thread.

* Removed HTTP class constants in `ThreadController`.

* Fixed form disappearing when creating a new reply.

* Fixed failing validator with encoded permalink.