/\*!CK:3324728596!\*/

/\*1415115209,\*/

if (self.CavalryLogger) {

CavalryLogger.start\_js(["18NMf"]);

}

\_\_d("PUWErrorCodes", [], function(a, b, c, d, e, f) {

e.exports = {

BAD\_JPEG: 9901,

WORKER\_TERMINATED: 9902,

UNKNOWN\_RESIZE\_ERROR: 9903,

WORKER\_ABORT: 9904

};

}, null);

\_\_d("StickerAssetType", [], function(a, b, c, d, e, f) {

e.exports = {

IMAGE: "BestEffortImage",

SPRITE: "SpriteImage",

PADDED\_SPRITE: "PaddedSpriteImage"

};

}, null);

\_\_d("TypingDetector", ["ArbiterMixin", "Event", "Input", "Run", "TypingStates", "copyProperties", "createObjectFrom", "emptyFunction"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n) {

function o(p, q) {

this.\_input = p;

this.\_ignoreKeys = {};

this.\_getValueFn = q;

}

l(o.prototype, g, {

\_timeout: 7000,

\_currentState: k.INACTIVE,

init: function() {

this.init = n;

this.reset();

h.listen(this.\_input, 'keyup', this.\_update.bind(this));

j.onUnload(this.\_onunload.bind(this));

},

reset: function() {

clearTimeout(this.\_checkTimer);

this.\_checkTimer = null;

this.\_lastKeystrokeAt = null;

this.\_currentState = k.INACTIVE;

},

setIgnoreKeys: function(p) {

this.\_ignoreKeys = m(p);

},

\_onunload: function() {

if (this.\_currentState == k.TYPING)

this.\_transition(k.QUITTING);

},

\_update: function(event) {

var p = h.getKeyCode(event), q = this.\_currentState;

if (!this.\_ignoreKeys[p]) {

var r = this.\_getValueFn ? this.\_getValueFn(): i.getValue(this.\_input);

if (r.trim().length === 0) {

if (q == k.TYPING)

this.\_transition(k.INACTIVE);

} else if (q == k.TYPING) {

this.\_recordKeystroke();

} else if (q == k.INACTIVE) {

this.\_transition(k.TYPING);

this.\_recordKeystroke();

}

}

},

\_transition: function(p) {

this.reset();

this.\_currentState = p;

this.inform('change', p);

},

\_recordKeystroke: function() {

this.\_lastKeystrokeTime = Date.now();

if (!this.\_checkTimer)

this.\_checkTimer = setTimeout(this.\_checkTyping.bind(this), this.\_timeout);

},

\_checkTyping: function() {

var p = this.\_lastKeystrokeTime + this.\_timeout, q = Date.now();

if (q > p) {

this.\_transition(k.INACTIVE);

} else {

clearTimeout(this.\_checkTimer);

this.\_checkTimer = setTimeout(this.\_checkTyping.bind(this), p - q + 10);

}

}

});

e.exports = o;

}, null);

\_\_d("TypingDetectorController", ["AsyncRequest", "AvailableList", "AvailableListConstants", "ChannelConnection", "ChatVisibility", "Keys", "PresencePrivacy", "ShortProfiles", "TypingDetector", "TypingStates", "copyProperties", "emptyFunction", "setTimeoutAcrossTransitions", "shield"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t) {

function u(w) {

if (!k.isOnline())

return false;

if (w) {

var x = n.getNow(w);

return x && x.type == "friend" && m.allows(w);

}

return true;

}

function v(w, x, y, z, aa, ba) {

"use strict";

this.userID = w;

this.input = x;

this.source = y;

this.threadID = aa;

this.remoteState = p.INACTIVE;

this.notifyTimer = null;

z = z || {};

this.notifyDelay = z.notifyDelay || this.notifyDelay;

this.\_typingDetector = new o(x, ba);

this.\_typingDetector.init(z);

this.\_typingDetector.subscribe('change', this.\_stateChange.bind(this));

}

v.prototype.setUserAndThread = function(w, x) {

"use strict";

if (this.userID !== w || this.threadID !== x) {

this.resetState();

this.userID = w;

this.threadID = x;

}

};

v.prototype.setIgnoreEnter = function(w) {

"use strict";

var x = w ? [l.RETURN]: [];

this.\_typingDetector.setIgnoreKeys(x);

};

v.prototype.resetState = function() {

"use strict";

this.\_notifyState(p.INACTIVE);

this.remoteState = p.INACTIVE;

this.\_typingDetector.reset();

clearTimeout(this.notifyTimer);

this.notifyTimer = null;

};

v.prototype.\_stateChange = function(w, x) {

"use strict";

if (x != p.QUITTING) {

clearTimeout(this.notifyTimer);

this.notifyTimer = s(t(this.\_notifyState, this, x), this.notifyDelay);

} else

this.\_notifyState(x, true);

};

v.prototype.\_notifyState = function(w, x) {

"use strict";

if (!this.userID&&!this.threadID)

return;

var y = this.userID, z = u(y);

if (z && w != this.remoteState) {

this.remoteState = w;

if (j.disconnected())

return;

var aa = {

typ: w,

to: y,

source: this.source,

thread: this.threadID

};

new g().setHandler(this.\_onTypResponse.bind(this, y)).setErrorHandler(r).setData(aa).setURI('/ajax/messaging/typ.php').setAllowCrossPageTransition(true).setOption('asynchronous', !x).send();

}

};

v.prototype.\_onTypResponse = function(w, x) {

"use strict";

var y = x.getPayload() || {};

if (y.offline)

h.set(w, i.OFFLINE, 'typing\_response');

};

q(v.prototype, {

notifyDelay: 1000

});

e.exports = v;

}, null);

\_\_d("VideoCallingTour", ["Arbiter", "ArbiterMixin", "Chat", "ChatSidebar", "ChatVisibility", "CSS", "DOM", "PresencePrivacy", "Run", "Toggler", "Vector", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r) {

var s, t, u, v, w = [], x = function() {};

function y() {

if (j.isVisible()) {

z();

} else if (u)

aa();

}

function z() {

s.setContext(t.getBody());

ba();

s.show();

ca();

}

function aa() {

if (!v)

v = p.createInstance(u.getRoot());

var fa = m.scry(u.getRoot(), 'div.fbNubFlyout')[0];

if (fa) {

s.setContext(fa);

ba();

s.show();

ca();

}

}

function ba() {

var fa = q.getElementDimensions(s.getContext()).y;

s.setOffsetY(fa \* .6);

s.updatePosition();

}

function ca() {

if (u)

w.push(u.subscribe('hide', function() {

da();

if (!j.isVisible())

s.hide();

}), u.subscribe('show', function() {

s.show();

}), u.subscribe('resize', function() {

ba();

s.updatePosition();

}));

w.push(g.subscribe('sidebar/show', z), g.subscribe('sidebar/hide', aa), g.subscribe('sidebar/resized', ba));

}

function da() {

if (v) {

v.setSticky(false);

v = null;

}

}

function ea() {

while (w.length)

w.pop().unsubscribe();

if (u)

da();

s.hide();

l.show('fbVideoCallingGetStarted');

}

r(x, h, {

start: function(fa) {

s = fa;

l.hide('fbVideoCallingGetStarted');

k.goOnline(function() {

w.push(n.subscribe('privacy-user-presence-changed', ea));

o.onLeave(ea);

i.openBuddyList();

var ga = null;

w.push(j.subscribe('sidebar/initialized', function(ha, ia) {

t = ia;

clearTimeout(ga);

ga = setTimeout(y, 0);

}), x.subscribe('videocallingtour/end', ea));

w.push(g.subscribe('buddylist-nub/initialized', function(ha, ia) {

u = ia;

clearTimeout(ga);

ga = setTimeout(y, 0);

}));

});

x.inform('videocallingtour/start');

}

});

e.exports = x;

}, null);

\_\_d("ContentRollTypeaheadView.react", ["BackgroundImage.react", "ImageBlock.react", "LegacyScrollableArea.react", "React", "TypeaheadViewItem", "cx", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l, m) {

var n = j.createClass({

displayName: 'ContentRollTypeaheadViewItem',

mixins: [k.Mixin],

propTypes: k.propTypes,

render: function() {

var p = this.props.entry, q = p.getSubtitle() ? j.createElement("div", {

className: "\_599q"

}, p.getSubtitle()): null;

if (q && p.getAuxiliaryData() && p.getAuxiliaryData().icon)

q = j.createElement(h, {

spacing: "small"

}, j.createElement(g, {

width: 16,

height: 16,

src: p.getAuxiliaryData().icon

}), q);

var r = p.getPhoto() ? j.createElement(g, {

width: 32,

height: 32,

backgroundSize: "cover",

src: p.getPhoto()

}): j.createElement("span", null), s = (("\_5yww") + (!q ? ' ' + "\_5mne" : '') + (this.props.highlighted ? ' ' + "\_599n" : ''));

return (j.createElement("li", {

className: s,

onMouseDown: this.\_onSelect,

onMouseEnter: this.\_onHighlight

}, j.createElement(h, {

spacing: "medium"

}, r, j.createElement("div", null, j.createElement("div", {

className: "\_599p"

}, p.getTitle()), q))));

}

}), o = j.createClass({

displayName: 'ContentRollTypeaheadView',

propTypes: {

highlightedEntry: j.PropTypes.object,

entries: j.PropTypes.array.isRequired,

onSelect: j.PropTypes.func.isRequired,

onHighlight: j.PropTypes.func,

onRenderHighlight: j.PropTypes.func,

scrollableAreaHeight: j.PropTypes.number.isRequired,

scrollableAreaWidth: j.PropTypes.number.isRequired

},

\_renderItem: function(p) {

var q = p === this.props.highlightedEntry;

return (j.createElement(n, {

key: p.getUniqueID(),

entry: p,

highlighted: q,

onSelect: this.props.onSelect,

onHighlight: this.props.onHighlight,

onRenderHighlight: this.props.onRenderHighlight

}));

},

render: function() {

var p = (("\_2072") + (!this.props.entries.length ? ' ' + "\_599s" : '')), q = this.props.entries, r = {};

q.forEach(function(v) {

r[v.getType()] = r[v.getType()] || [];

r[v.getType()].push(v);

});

var s = [];

for (var t in r) {

var u = null;

switch (t) {

case 'recent':

u = "Recently Viewed Links";

break;

case 'saved':

u = "Saved Links";

break;

case 'bing':

u = "Bing Results";

break;

}

if (u)

s.push(j.createElement("div", {

className: "\_1opo"

}, u));

r[t].forEach(function(v) {

s.push(this.\_renderItem(v));

}, this);

}

return (j.createElement("div", {

className: p

}, j.createElement(i, {

height: this.props.scrollableAreaHeight,

width: this.props.scrollableAreaWidth

}, j.createElement("ul", null, s))));

}

});

e.exports = o;

}, null);

\_\_d("ContentRollTypeahead.react", ["AbstractTypeahead.react", "React", "ReactPropTypes", "WebAsyncSearchSource", "ContentRollTypeaheadView.react", "cx", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l, m) {

var n = 20, o = '/ajax/metacomposer/attachment/og/typeahead/content\_roll', p = h.createClass({

displayName: 'ContentRollTypeahead',

propTypes: {

className: i.string,

onSelection: i.func.isRequired,

scrollableAreaHeight: h.PropTypes.number.isRequired,

scrollableAreaWidth: h.PropTypes.number.isRequired,

userID: i.string.isRequired,

useLayer: i.bool

},

getInitialState: function() {

return {

query: ''

};

},

\_handleChange: function(event) {

this.setState({

query: event.target.value

});

},

\_handleSelectAttempt: function(q) {

this.setState({

query: ''

});

var r = {

object\_id: q.getUniqueID(),

url: q.getURI(),

source: q.getType(),

query: q.getAuxiliaryData().query,

subindex: q.getAuxiliaryData().subindex,

subtotal: q.getAuxiliaryData().subtotal,

index: q.getAuxiliaryData().index,

total: q.getAuxiliaryData().total

};

this.props.onSelection(r);

},

\_getSearchSource: function() {

if (!this.\_searchSource)

this.\_searchSource = new j({

bootstrapRequests: [{

uri: o,

data: {

viewer: this.props.userID

}

}

],

queryRequests: [{

uri: o,

data: {

viewer: this.props.userID

}

}

],

getAllForEmptyQuery: true

});

return this.\_searchSource;

},

\_getPlaceholder: function() {

if (!this.\_placeholder)

this.\_placeholder = "Search for articles, videos, music, and more";

return this.\_placeholder;

},

focusInput: function() {

this.refs.typeahead.focusInput();

},

render: function() {

var q = {

ViewRenderer: k,

maxEntries: n,

useLayer: this.props.useLayer,

extraRendererProps: {

scrollableAreaHeight: this.props.scrollableAreaHeight,

scrollableAreaWidth: this.props.scrollableAreaWidth

}

};

return (h.createElement(g, {

autoHighlight: true,

className: this.props.className,

focusedOnInit: true,

inputClassName: "\_55r1",

onChange: this.\_handleChange,

onSelectAttempt: this.\_handleSelectAttempt,

placeholder: this.\_getPlaceholder(),

presenter: q,

queryString: this.state.query,

ref: "typeahead",

searchSource: this.\_getSearchSource(),

showEntriesOnFocus: true

}));

}

});

e.exports = p;

}, null);

\_\_d("ContentRollFlyout.react", ["ContentRollTypeahead.react", "Parent", "React", "ReactPropTypes", "Toggler", "cx"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = i.createClass({

displayName: 'ContentRollFlyout',

propTypes: {

onSelection: j.func.isRequired,

userID: j.string.isRequired

},

getInitialState: function() {

return {

showFlyout: false

};

},

componentDidMount: function() {

var n = this.\_getParentToggler();

if (n)

k.listen('show', n, function() {

this.setState({

showFlyout: true

});

}.bind(this));

},

\_getParentToggler: function() {

return h.byClass(this.getDOMNode(), 'uiToggle');

},

\_hideParentToggler: function() {

var n = this.\_getParentToggler();

if (n)

k.hide(n);

},

\_handleSelection: function(n) {

this.\_hideParentToggler();

this.props.onSelection(n);

},

render: function() {

if (!this.state.showFlyout)

return i.createElement("div", null);

return (i.createElement(g, {

className: "\_5j11",

onSelection: this.\_handleSelection,

scrollableAreaHeight: 288,

scrollableAreaWidth: 256,

userID: this.props.userID,

useLayer: false

}));

}

});

e.exports = m;

}, null);

\_\_d("CroppedImage.react", ["Image.react", "React", "ReactPropTypes", "cx"], function(a, b, c, d, e, f, g, h, i, j) {

var k = h.createClass({

displayName: 'CroppedImage',

propTypes: {

center: i.shape({

x: i.number.isRequired,

y: i.number.isRequired

}),

crop: i.shape({

height: i.number.isRequired,

width: i.number.isRequired

}).isRequired,

size: i.shape({

height: i.number.isRequired,

width: i.number.isRequired

}).isRequired,

src: i.string.isRequired

},

getDefaultProps: function() {

return {

center: {

x: .5,

y: .5

}

};

},

\_calculateMargin: function(l, m, n) {

return - 1 \* Math.max(Math.min(((n \* m) - (.5 \* l)), m - l), 0);

},

render: function() {

var l = this.\_calculateMargin(this.props.crop.width, this.props.size.width, this.props.center.x), m = this.\_calculateMargin(this.props.crop.height, this.props.size.height, this.props.center.y), n = {

left: l,

height: this.props.size.height,

top: m,

width: this.props.size.width

};

return (h.createElement("div", {

className: "\_46-h",

style: this.props.crop

}, h.createElement(g, {

alt: this.props.alt || '',

className: "\_46-i",

src: this.props.src,

style: n

})));

}

});

e.exports = k;

}, null);

\_\_d("URLScraper", ["ArbiterMixin", "DataStore", "Event", "URLMatcher", "copyProperties", "mixin"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = 'scraperLastPermissiveMatch', n = l(g);

for (var o in n)

if (n.hasOwnProperty(o))

q[o] = n[o];

var p = n === null ? null: n.prototype;

q.prototype = Object.create(p);

q.prototype.constructor = q;

q.\_\_superConstructor\_\_ = n;

function q(r, s) {

"use strict";

this.input = r;

this.enable();

this.getValueFn = s;

}

q.prototype.reset = function() {

"use strict";

h.set(this.input, m, null);

};

q.prototype.enable = function() {

"use strict";

if (this.events)

return;

var r = function(s) {

setTimeout(this.check.bind(this, s), 30);

};

this.events = i.listen(this.input, {

paste: r.bind(this, false),

keydown: r.bind(this, true)

});

};

q.prototype.disable = function() {

"use strict";

if (!this.events)

return;

for (var event in this.events)

this.events[event].remove();

this.events = null;

};

q.prototype.check = function(r) {

"use strict";

var s = this.getValueFn ? this.getValueFn(): this.input.value;

if (r && q.trigger(s))

return;

var t = q.match(s), u = j.permissiveMatch(s);

if (u && (u != h.get(this.input, m))) {

h.set(this.input, m, u);

this.inform('match', {

url: t || u,

alt\_url: u

});

}

};

k(q, j);

e.exports = q;

}, null);

\_\_d("getURLRanges", ["URI", "URLScraper"], function(a, b, c, d, e, f, g, h) {

"use strict";

function i(j, k) {

k = k || 0;

var l = h.match(j);

if (!l)

return [];

var m = l;

if (!(/^[a-z][a-z0-9\-+.]+:\/\//i.test(l)))

m = 'http://' + l;

if (!g.isValidURI(m))

return [];

k += j.indexOf(l);

var n = l.length, o = [{

offset: k,

length: l.length,

entity: {

url: m

}

}

];

return o.concat(i(j.substr(k + n), k + n));

}

e.exports = i;

}, null);

\_\_d("MercuryMessageBody.react", ["React", "Link.react", "ReactComponentWithPureRenderMixin", "TextWithEntities.react", "cx", "fbt", "getURLRanges"], function(a, b, c, d, e, f, g, h, i, j, k, l, m) {

var n = function(p, q) {

return (g.createElement(h, {

target: "\_blank",

href: q.entity

}, p));

}, o = g.createClass({

displayName: 'MercuryMessageBody',

mixins: [i],

propTypes: {

body: g.PropTypes.string,

ranges: g.PropTypes.array

},

render: function() {

var p = this.props.body || '';

if (p.startsWith('?OTR'))

return (g.createElement("span", {

className: "\_89h"

}, "[verschl\u00fcsselte Nachricht]"));

p = p.replace(/\s+$/, '');

if (p.length === 0)

return null;

var q = this.props.ranges;

if (q === undefined)

q = m(p);

return (g.createElement(j, {

interpolator: n,

text: p,

ranges: q,

renderEmoticons: true,

renderEmoji: true

}));

}

});

e.exports = o;

}, null);

\_\_d("MercuryMessageError.react", ["CurrentUser", "ImmutableObject", "MercuryErrorInfo", "MercuryMessages", "React", "cx", "joinClasses"], function(a, b, c, d, e, f, g, h, i, j, k, l, m) {

'use strict';

var n = k.PropTypes, o = k.createClass({

displayName: 'MercuryMessageError',

propTypes: {

authorFBID: n.string,

message: n.instanceOf(h).isRequired

},

getDefaultProps: function() {

return {

authorFBID: g.getID()

};

},

render: function() {

var p = this.props.message, q = p.error\_data;

return (k.createElement("div", Object.assign({}, this.props, {

className: m(this.props.className, this.\_getClassNameFromStatus(p)),

tabIndex: i.isTransient(q) ? 0: null,

onClick: i.isTransient(q) ? this.messageResend: null

}), k.createElement("div", {

className: "\_2fs1"

}), k.createElement("span", {

dangerouslySetInnerHTML: {

\_\_html: this.\_getTextFromStatus(p)

}

})));

},

messageResend: function() {

j.getForFBID(this.props.authorFBID).resendMessage(this.props.message);

},

\_getClassNameFromStatus: function(p) {

if (i.hasErrorStatus(p)) {

return (("\_2fs2") + (' ' + "\_2fs3") + (i.isTransient(p.error\_data) ? ' ' + "\_2fs4" : '') + (i.isPermanent(p.error\_data) ? ' ' + "\_2fs5" : ''));

} else

return "\_2fs6 \_2fs2";

},

\_getTextFromStatus: function(p) {

var q = '', r = p.error\_data;

if (i.hasErrorStatus(p))

q = i.getMessage(r);

if (typeof q === 'object' && q.\_\_html)

q = q.\_\_html;

return q;

}

});

e.exports = o;

}, null);

\_\_d("MercuryAudioPlayer", ["Event", "Arbiter", "DOM", "Flash", "UserAgent\_DEPRECATED", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = 200;

function n() {

if (k.webkit()&&!k.chrome())

return false;

var w = i.create('audio'), x = false;

try {

if (!!w.canPlayType)

if (w.canPlayType('video/mp4;').replace(/^no$/, ''))

x = true;

} finally {

return x;

}

}

function o() {

return j.isAvailable();

}

var p = function() {

this.interval = null;

this.arbiterInstance = null;

this.audio = i.create('audio');

g.listen(this.audio, 'playing', function() {

this.informAttachment('playing', this.audio.currentTime);

this.interval = setInterval(function() {

this.informAttachment('playing', this.audio.currentTime);

}.bind(this), m);

}.bind(this));

g.listen(this.audio, 'ended', function() {

clearInterval(this.interval);

this.informAttachment('finished');

}.bind(this));

};

l(p.prototype, {

setAudio: function(w, x) {

this.audio.setAttribute('src', w);

this.arbiterInstance = x;

},

informAttachment: function(w, x) {

if (this.arbiterInstance)

this.arbiterInstance.inform(w, x);

},

play: function() {

this.audio.play();

this.informAttachment('played');

},

resume: function() {

this.audio.play();

this.informAttachment('played');

},

pause: function() {

this.audio.pause();

clearInterval(this.interval);

this.informAttachment('paused');

},

getType: function() {

return 'html5';

}

});

var q = function() {

this.src = null;

this.arbiterInstance = null;

var w = i.create('div');

document.body.appendChild(w);

this.swf = j.embed('/swf/SoundStreamPlayer.swf', w, null, {});

this.interval = null;

h.subscribe('soundstream/finished', function() {

clearInterval(this.interval);

this.informAttachment('finished');

}.bind(this));

};

l(q.prototype, {

setAudio: function(w, x) {

this.src = w;

this.arbiterInstance = x;

},

informAttachment: function(w, x) {

if (this.arbiterInstance)

this.arbiterInstance.inform(w, x);

},

play: function() {

this.swf.playSound(this.src);

this.interval = setInterval(function() {

var w = this.swf.getCurrentTime();

this.informAttachment('playing', w);

}.bind(this), m);

this.informAttachment('played');

},

resume: function() {

this.swf.resume();

this.informAttachment('played');

},

pause: function() {

clearInterval(this.interval);

this.swf.pause();

this.informAttachment('paused');

},

getType: function() {

return 'flash';

}

});

function r() {

if (n()) {

return new p();

} else if (o())

return new q();

return false;

}

var s = null, t = null, u = 0;

function v(w, x) {

this.src = w;

this.arbiterInstance = x;

this.audio\_id=++u;

s !== null || (s = r());

if (!s)

return false;

}

l(v.prototype, {

getType: function() {

if (!s) {

return false;

} else

return s.getType();

},

play: function(w) {

if (w && t == this.audio\_id) {

s.resume();

} else {

this.pause();

t = this.audio\_id;

s.setAudio(this.src, this.arbiterInstance);

s.play();

}

},

pause: function() {

s.pause();

}

});

e.exports = v;

}, null);

\_\_d("MercuryAttachmentAudioClip.react", ["Arbiter", "ArbiterMixin", "MercuryAudioPlayer", "CurrentUser", "JSLogger", "LeftRight.react", "React", "cx", "shield", "tx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

var q = 'MercuryAttachmentAudioClip/play', r = k.create('mercury\_audio\_clip'), s = m.createClass({

displayName: 'AudioClip',

mixins: [h],

getInitialState: function() {

this.subscribe('playing', this.updateTime);

this.subscribe('played', o(this.setState, this, {

playing: true,

started: true

}));

this.subscribe('paused', o(this.setState, this, {

playing: false

}));

this.subscribe('finished', o(this.setState, this, {

playing: false,

started: false,

time: this.props.duration

}));

this.logged = false;

var t = this.props.downloadOnly ? false: new i(this.props.src, this);

g.subscribe(q, function(u, v) {

if (this.props.src != v)

this.setState({

time: 0

});

}.bind(this));

return {

time: 0,

playing: false,

started: false,

duration: this.props.duration,

audioPlayer: t

};

},

updateTime: function(t, u) {

this.setState({

time: u

});

},

play: function() {

if (this.state.playing) {

this.state.audioPlayer.pause();

} else {

this.state.audioPlayer.play(this.state.started);

g.inform(q, this.props.src);

if (!this.logged) {

this.logged = true;

r.log('play', {

uid: j.getID(),

duration: this.props.duration

});

}

}

},

\_formatSeconds: function(t) {

if (t) {

t = Math.ceil(t);

var u = t%60;

if (u < 10)

u = '0' + u;

var v = Math.floor(t / 60);

return v + ':' + u;

} else

return null;

},

\_renderPlayer: function(t, u) {

return (m.createElement("a", {

className: "\_1miz",

href: "#",

style: {

width: t

},

onClick: this.play

}, m.createElement("span", {

className: "\_1mi-"

}, m.createElement("i", {

className: "\_1mi\_"

})), m.createElement("span", {

className: "\_1mj0"

}, u), m.createElement("div", {

className: "\_1mj1"

})));

},

render: function() {

var t = this.state.time, u = this.state.playing, v = this.\_formatSeconds(this.state.duration), w = this.props.width || 170, x = null, y = Math.ceil((t \* (w + 2)) / this.state.duration);

if (this.state.audioPlayer && this.state.audioPlayer.getType()) {

var z = this.\_renderPlayer(w, v), aa = this.\_renderPlayer(w, v), ba = (("\_1mj2") + (u && (t !== 0) ? ' ' + "\_1mj3" : '') + (u && (t === 0) ? ' ' + "\_4g4x" : ''));

x = (m.createElement("div", {

className: ba

}, z, m.createElement("div", {

className: "\_1mj4",

style: {

width: y

}

}, aa)));

} else

x = (m.createElement("div", {

className: "\_1mj2"

}, m.createElement("div", {

className: "\_1miz"

}, m.createElement(l, null, m.createElement("a", {

className: "\_1mj5",

href: this.props.src

}, m.createElement("span", {

className: "\_3qi6"

}, m.createElement("i", {

className: "\_1mj6"

})), m.createElement("span", {

className: "\_1mj7"

}, "Sprachnachricht"), m.createElement("span", {

className: "\_1mj8"

}, v)), m.createElement("a", {

href: this.props.src,

className: "\_1mj9"

}, m.createElement("i", {

className: "\_1mja"

}))))));

return (m.createElement("div", {

className: "\_1mjb"

}, x));

}

});

e.exports = s;

}, null);

\_\_d("XMessagingVideoAttachmentControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/mercury\/attachments\/video\/", {

video\_id: {

type: "Int",

required: true

},

window\_width: {

type: "Int",

required: true

},

window\_height: {

type: "Int",

required: true

}

});

}, null);

\_\_d("MercuryAttachmentVideo.react", ["ActorURI", "CroppedImage.react", "Image.react", "React", "ReactPropTypes", "XMessagingVideoAttachmentControllerURIBuilder", "cx", "getViewportDimensions", "ix"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o) {

var p = j.createClass({

displayName: 'MercuryAttachmentVideo',

propTypes: {

duration: k.number.isRequired,

name: k.string.isRequired,

pageID: k.number,

size: k.shape({

height: k.number,

width: k.number

}),

thumbnailSize: k.shape({

height: k.number,

width: k.number

}),

thumbnail: k.string,

videoID: k.string

},

\_formatDuration: function() {

var q = this.props.duration, r = Math.floor(q / 60), s = q%60;

if (s < 10)

return r + ':0' + s;

return r + ':' + s;

},

render: function() {

var q=!!this.props.thumbnail, r = (("\_zow") + (' ' + "\_59go") + (!q ? ' ' + "\_59gq" : '')), s = ((!q ? "\_3l6h" : '') + (!q ? ' ' + "uiIconText" : '')), t = n(), u = new l().setInt('video\_id', this.props.videoID).setInt('window\_width', t.width).setInt('window\_height', t.height).getURI();

if (this.props.pageID)

u = g.create(u, this.props.pageID);

if (this.props.thumbnail)

return (j.createElement("div", {

className: r,

style: this.props.size

}, j.createElement("a", {

className: s,

href: u,

role: "button",

rel: "async"

}, j.createElement(h, {

crop: this.props.size,

size: this.props.thumbnailSize,

src: this.props.thumbnail

}), j.createElement("div", {

className: "\_zox"

}), j.createElement("div", {

className: "\_zoy"

}, j.createElement(i, {

src: o('/images/chat/chat\_play\_icon.png')

})), j.createElement("span", {

className: "\_zoz"

}, this.\_formatDuration()))));

return (j.createElement("div", {

className: r

}, j.createElement("a", {

className: s,

href: u,

role: "button",

rel: "async"

}, j.createElement(i, {

src: o('/images/icons/video.gif')

}), j.createElement("span", {

className: "\_59gp"

}, this.props.name))));

}

});

e.exports = p;

}, null);

\_\_d("XUIAmbientNUX.react", ["HasLayerContextMixin", "React", "ReactLayer", "ReactAbstractContextualDialog", "XUIAmbientNUXTheme", "XUICloseButton.react", "XUIContextualDialogBody.react", "cx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n) {

var o = h.PropTypes, p = 300, q = 380, r = i.createClass(j.createSpec({

displayName: 'XUIAmbientNUX',

theme: k

})), s = h.createClass({

displayName: 'XUIAmbientNUX',

mixins: [g],

propTypes: {

alignment: o.oneOf(['left', 'center', 'right']),

behaviors: o.object,

context: o.object,

contextRef: o.string,

customwidth: o.number,

offsetX: o.number,

offsetY: o.number,

onCloseButtonClick: o.func,

position: o.oneOf(['above', 'below', 'left', 'right']),

shown: o.bool,

width: o.oneOf(['wide', 'normal', 'auto', 'custom'])

},

\_getWidth: function() {

switch (this.props.width) {

case 'wide':

return q;

case 'custom':

return this.props.customwidth;

case 'auto':

return null;

default:

return p;

}

},

render: function() {

return (h.createElement(r, {

alignment: this.props.alignment,

autoFocus: false,

behaviors: this.props.behaviors,

context: this.getContextNode(),

focusContextOnHide: false,

offsetX: this.props.offsetX,

offsetY: this.props.offsetY,

position: this.props.position,

shown: this.props.shown,

width: this.\_getWidth(this.props)

}, h.createElement("div", {

className: "\_53iv"

}, h.createElement(m, null, h.createElement(l, {

shade: "light",

className: "\_36gl",

onClick: this.props.onCloseButtonClick

}), h.createElement("div", {

className: "\_36gn"

}, this.props.children)))));

}

});

e.exports = s;

}, null);

\_\_d("updatePhotoProgressBar", [], function(a, b, c, d, e, f) {

'use strict';

function g(h, event) {

if (event.loaded != event.total) {

h.setPosition(50 \* event.loaded / event.total);

} else {

h.setPosition(50);

h.setTarget(100, 2000);

}

}

e.exports = g;

}, null);

\_\_d("md5", ["str2rstr"], function(a, b, c, d, e, f, g) {

function h(u, v) {

var w = u[0], x = u[1], y = u[2], z = u[3];

w = j(w, x, y, z, v[0], 7, - 680876936);

z = j(z, w, x, y, v[1], 12, - 389564586);

y = j(y, z, w, x, v[2], 17, 606105819);

x = j(x, y, z, w, v[3], 22, - 1044525330);

w = j(w, x, y, z, v[4], 7, - 176418897);

z = j(z, w, x, y, v[5], 12, 1200080426);

y = j(y, z, w, x, v[6], 17, - 1473231341);

x = j(x, y, z, w, v[7], 22, - 45705983);

w = j(w, x, y, z, v[8], 7, 1770035416);

z = j(z, w, x, y, v[9], 12, - 1958414417);

y = j(y, z, w, x, v[10], 17, - 42063);

x = j(x, y, z, w, v[11], 22, - 1990404162);

w = j(w, x, y, z, v[12], 7, 1804603682);

z = j(z, w, x, y, v[13], 12, - 40341101);

y = j(y, z, w, x, v[14], 17, - 1502002290);

x = j(x, y, z, w, v[15], 22, 1236535329);

w = k(w, x, y, z, v[1], 5, - 165796510);

z = k(z, w, x, y, v[6], 9, - 1069501632);

y = k(y, z, w, x, v[11], 14, 643717713);

x = k(x, y, z, w, v[0], 20, - 373897302);

w = k(w, x, y, z, v[5], 5, - 701558691);

z = k(z, w, x, y, v[10], 9, 38016083);

y = k(y, z, w, x, v[15], 14, - 660478335);

x = k(x, y, z, w, v[4], 20, - 405537848);

w = k(w, x, y, z, v[9], 5, 568446438);

z = k(z, w, x, y, v[14], 9, - 1019803690);

y = k(y, z, w, x, v[3], 14, - 187363961);

x = k(x, y, z, w, v[8], 20, 1163531501);

w = k(w, x, y, z, v[13], 5, - 1444681467);

z = k(z, w, x, y, v[2], 9, - 51403784);

y = k(y, z, w, x, v[7], 14, 1735328473);

x = k(x, y, z, w, v[12], 20, - 1926607734);

w = l(w, x, y, z, v[5], 4, - 378558);

z = l(z, w, x, y, v[8], 11, - 2022574463);

y = l(y, z, w, x, v[11], 16, 1839030562);

x = l(x, y, z, w, v[14], 23, - 35309556);

w = l(w, x, y, z, v[1], 4, - 1530992060);

z = l(z, w, x, y, v[4], 11, 1272893353);

y = l(y, z, w, x, v[7], 16, - 155497632);

x = l(x, y, z, w, v[10], 23, - 1094730640);

w = l(w, x, y, z, v[13], 4, 681279174);

z = l(z, w, x, y, v[0], 11, - 358537222);

y = l(y, z, w, x, v[3], 16, - 722521979);

x = l(x, y, z, w, v[6], 23, 76029189);

w = l(w, x, y, z, v[9], 4, - 640364487);

z = l(z, w, x, y, v[12], 11, - 421815835);

y = l(y, z, w, x, v[15], 16, 530742520);

x = l(x, y, z, w, v[2], 23, - 995338651);

w = m(w, x, y, z, v[0], 6, - 198630844);

z = m(z, w, x, y, v[7], 10, 1126891415);

y = m(y, z, w, x, v[14], 15, - 1416354905);

x = m(x, y, z, w, v[5], 21, - 57434055);

w = m(w, x, y, z, v[12], 6, 1700485571);

z = m(z, w, x, y, v[3], 10, - 1894986606);

y = m(y, z, w, x, v[10], 15, - 1051523);

x = m(x, y, z, w, v[1], 21, - 2054922799);

w = m(w, x, y, z, v[8], 6, 1873313359);

z = m(z, w, x, y, v[15], 10, - 30611744);

y = m(y, z, w, x, v[6], 15, - 1560198380);

x = m(x, y, z, w, v[13], 21, 1309151649);

w = m(w, x, y, z, v[4], 6, - 145523070);

z = m(z, w, x, y, v[11], 10, - 1120210379);

y = m(y, z, w, x, v[2], 15, 718787259);

x = m(x, y, z, w, v[9], 21, - 343485551);

u[0] = s(w, u[0]);

u[1] = s(x, u[1]);

u[2] = s(y, u[2]);

u[3] = s(z, u[3]);

}

function i(u, v, w, x, y, z) {

v = s(s(v, u), s(x, z));

return s((v<<y) | (v>>>(32 - y)), w);

}

function j(u, v, w, x, y, z, aa) {

return i((v & w) | ((~v) & x), u, v, y, z, aa);

}

function k(u, v, w, x, y, z, aa) {

return i((v & x) | (w & (~x)), u, v, y, z, aa);

}

function l(u, v, w, x, y, z, aa) {

return i(v^w^x, u, v, y, z, aa);

}

function m(u, v, w, x, y, z, aa) {

return i(w^(v | (~x)), u, v, y, z, aa);

}

function n(u) {

var v = u.length, w = [1732584193, - 271733879, - 1732584194, 271733878], x;

for (x = 64; x <= u.length; x += 64)

h(w, o(u.substring(x - 64, x)));

u = u.substring(x - 64);

var y = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

for (x = 0; x < u.length; x++)

y[x>>2]|=u.charCodeAt(x)<<((x & 3)<<3);

y[x>>2]|=128<<((x & 3)<<3);

if (x > 55) {

h(w, y);

for (x = 0; x < 16; x++)

y[x] = 0;

}

y[14] = v \* 8;

h(w, y);

return w;

}

function o(u) {

var v = [], w = 0;

while (w < 64)

v[w>>2] = u.charCodeAt(w++) | (u.charCodeAt(w++)<<8) | (u.charCodeAt(w++)<<16) | (u.charCodeAt(w++)<<24);

return v;

}

var p = '0123456789abcdef'.split('');

function q(u) {

var v = '', w = 0;

for (; w < 4; w++)

v += p[(u>>((w<<3) + 4)) & 15] + p[(u>>(w<<3)) & 15];

return v;

}

function r(u) {

for (var v = 0; v < u.length; v++)

u[v] = q(u[v]);

return u.join('');

}

var s = function(u, v) {

return (u + v) & 4294967295;

};

function t(u) {

if (null === u || undefined === u) {

return null;

} else {

for (var v = 0; v < u.length; v++)

if (u[v] > "\u007F") {

u = g(u);

break;

}

return r(n(u));

}

}

if (t('hello') != '5d41402abc4b2a76b9719d911017c592')

s = function(u, v) {

var w = (u & 65535) + (v & 65535), x = (u>>16) + (v>>16) + (w>>16);

return (x<<16) | (w & 65535);

};

e.exports = t;

}, null);

\_\_d("DataViewPolyfill", [], function(a, b, c, d, e, f) {

"use strict";

function g(h, i, j) {

if (i === undefined) {

this.$DataViewPolyfill0 = new Uint8Array(h);

} else if (j === undefined) {

this.$DataViewPolyfill0 = new Uint8Array(h, i);

} else

this.$DataViewPolyfill0 = new Uint8Array(h, i, j);

this.byteLength = this.$DataViewPolyfill0.byteLength;

}

g.prototype.getUint8 = function(h) {

if (h >= this.$DataViewPolyfill0.length)

throw new Error('Trying to read beyond bounds of DataViewPolyfill');

return this.$DataViewPolyfill0[h];

};

g.prototype.getUint16 = function(h, i) {

var j = this.getUint8(h), k = this.getUint8(h + 1);

return i ? (k \* 256) + j : (j \* 256) + k;

};

g.prototype.getUint32 = function(h, i) {

var j = this.getUint8(h), k = this.getUint8(h + 1), l = this.getUint8(h + 2), m = this.getUint8(h + 3);

return i ? (((m \* 256 + l) \* 256 + k) \* 256) + j : (((j \* 256 + k) \* 256 + l) \* 256) + m;

};

g.isSupported = function() {

return !!a.Uint8Array;

};

e.exports = g;

}, null);

\_\_d("Emscripten", [], function(a, b, c, d, e, f) {

e.exports = {

isSupported: function() {

return (typeof Int32Array !== 'undefined' && typeof Float64Array !== 'undefined'&&!!new Int32Array(1).subarray&&!!new Int32Array(1).set);

}

};

}, null);

\_\_d("XHRHttpError", [], function(a, b, c, d, e, f) {

var g = 'HTTP\_CLIENT\_ERROR', h = 'HTTP\_PROXY\_ERROR', i = 'HTTP\_SERVER\_ERROR', j = 'HTTP\_TRANSPORT\_ERROR', k = 'HTTP\_UNKNOWN\_ERROR';

function l(m, n) {

if (n === 0) {

var o = m.getProtocol();

if (o === 'file' || o === 'ftp')

return null;

return j;

} else if (n >= 100 && n < 200) {

return h;

} else if (n >= 200 && n < 300) {

return null;

} else if (n >= 400 && n < 500) {

return g;

} else if (n >= 500 && n < 600) {

return i;

} else if (n >= 12001 && n < 12156) {

return j;

} else

return k;

}

e.exports = {

getErrorCode: l,

HTTP\_CLIENT\_ERROR: g,

HTTP\_PROXY\_ERROR: h,

HTTP\_SERVER\_ERROR: i,

HTTP\_TRANSPORT\_ERROR: j,

HTTP\_UNKNOWN\_ERROR: k

};

}, null);

\_\_d("XHRRequest", ["ErrorUtils", "URI", "XHRHttpError", "getSameOriginTransport", "invariant", "xhrSimpleDataSerializer"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = {

errorCode: null,

errorMsg: null,

errorType: null

}, n = {

loadedBytes: null,

totalBytes: null

};

function o(p) {

"use strict";

this.setURI(p);

this.setMethod('POST');

this.setIsAsynchronous(true);

this.setTransportBuilder(j);

this.setDataSerializer(l);

this.$XHRRequest0 = this.$XHRRequest0.bind(this);

}

o.prototype.setURI = function(p) {

"use strict";

this.$XHRRequest1 = new h(p);

return this;

};

o.prototype.getURI = function() {

"use strict";

return this.$XHRRequest1;

};

o.prototype.setMethod = function(p) {

"use strict";

this.$XHRRequest2 = p;

return this;

};

o.prototype.getMethod = function() {

"use strict";

return this.$XHRRequest2;

};

o.prototype.setData = function(p) {

"use strict";

this.$XHRRequest3 = p;

return this;

};

o.prototype.getData = function() {

"use strict";

return this.$XHRRequest3;

};

o.prototype.setRawData = function(p) {

"use strict";

this.$XHRRequest4 = p;

return this;

};

o.prototype.setRequestHeader = function(p, q) {

"use strict";

if (!this.$XHRRequest5)

this.$XHRRequest5 = {};

this.$XHRRequest5[p] = q;

return this;

};

o.prototype.setTimeout = function(p) {

"use strict";

this.$XHRRequest6 = p;

return this;

};

o.prototype.setResponseHandler = function(p) {

"use strict";

this.$XHRRequest7 = p;

return this;

};

o.prototype.setErrorHandler = function(p) {

"use strict";

this.$XHRRequest8 = p;

return this;

};

o.prototype.setAbortHandler = function(p) {

"use strict";

this.$XHRRequest9 = p;

return this;

};

o.prototype.setTimeoutHandler = function(p) {

"use strict";

this.$XHRRequesta = p;

return this;

};

o.prototype.setUploadProgressHandler = function(p) {

"use strict";

this.$XHRRequestb = p;

return this;

};

o.prototype.setIsAsynchronous = function(p) {

"use strict";

this.$XHRRequestc = p;

return this;

};

o.prototype.setTransportBuilder = function(p) {

"use strict";

this.$XHRRequestd = p;

return this;

};

o.prototype.setDataSerializer = function(p) {

"use strict";

this.$XHRRequeste = p;

return this;

};

o.prototype.send = function() {

"use strict";

var p = this.$XHRRequest6, q = this.$XHRRequestd(), r = this.getURI();

this.$XHRRequestf = q;

var s;

k(this.$XHRRequest2 === 'POST' ||!this.$XHRRequest4);

if (this.$XHRRequest2 === 'GET' || this.$XHRRequest4) {

r.addQueryData(this.$XHRRequest3);

s = this.$XHRRequest4;

} else

s = this.$XHRRequeste(this.$XHRRequest3);

q.onreadystatechange = this.$XHRRequest0;

if (q.upload && this.$XHRRequestb)

q.upload.onprogress = this.$XHRRequestg.bind(this);

if (p)

this.$XHRRequesth = setTimeout(this.$XHRRequesti.bind(this), p);

q.open(this.$XHRRequest2, r.toString(), this.$XHRRequestc);

if (this.$XHRRequest5)

for (var t in this.$XHRRequest5)

q.setRequestHeader(t, this.$XHRRequest5[t]);

q.send(s);

if (!this.$XHRRequestc)

q.onreadystatechange(null, true);

};

o.prototype.abort = function() {

"use strict";

this.$XHRRequestj();

if (this.$XHRRequest9)

g.applyWithGuard(this.$XHRRequest9, null, null, null, 'XHRRequest:\_abortHandler');

};

o.prototype.$XHRRequestj = function() {

"use strict";

var p = this.$XHRRequestf;

if (p) {

p.onreadystatechange = null;

p.abort();

}

this.$XHRRequestk();

};

o.prototype.$XHRRequesti = function() {

"use strict";

this.$XHRRequestj();

if (this.$XHRRequesta)

g.applyWithGuard(this.$XHRRequesta, null, null, null, 'XHRRequest:\_abortHandler');

};

o.prototype.$XHRRequest0 = function(p, q) {

"use strict";

if (!this.$XHRRequestc && q !== true)

return;

var r = this.$XHRRequestf, s = r, t = s.readyState;

if (t >= 2) {

var u = t === 4, v = this.getURI(), w = i.getErrorCode(v, r.status), x = this.$XHRRequest7;

if (w !== null) {

if (u) {

m.errorCode = w;

m.errorMsg = r.responseText;

m.errorType = 'HTTP';

if (this.$XHRRequest8)

g.applyWithGuard(this.$XHRRequest8, null, [m], null, 'XHRRequest:\_errorHandler');

}

} else if (x) {

var y = null;

if (x.includeHeaders)

y = r.getAllResponseHeaders();

if (u || (x.parseStreaming && t === 3))

g.applyWithGuard(x, null, [r.responseText, y, u], null, 'XHRRequest:handler');

}

if (u)

this.$XHRRequestk();

}

};

o.prototype.$XHRRequestg = function(p) {

"use strict";

n.loadedBytes = p.loaded;

n.totalBytes = p.total;

if (this.$XHRRequestb)

g.applyWithGuard(this.$XHRRequestb, null, [n], null, 'XHRRequest:\_uploadProgressHandler');

};

o.prototype.$XHRRequestk = function() {

"use strict";

clearTimeout(this.$XHRRequesth);

delete this.$XHRRequestf;

};

e.exports = o;

}, null);

\_\_d("getCrossOriginTransport", ["ex", "invariant"], function(a, b, c, d, e, f, g, h) {

function i() {

try {

var k = new XMLHttpRequest();

if (!('withCredentials' in k) && typeof XDomainRequest !== 'undefined')

k = new XDomainRequest();

return k;

} catch (j) {

throw new Error(g('getCrossOriginTransport: %s', j.message));

}

}

i.withCredentials = function() {

var j = i();

h('withCredentials' in j);

var k = j.open;

j.open = function() {

k.apply(this, arguments);

this.withCredentials = true;

};

return j;

};

e.exports = i;

}, null);

\_\_d("WebWorkerPool", ["PooledWebWorker", "emptyFunction"], function(a, b, c, d, e, f, g, h) {

i.isSupported = function() {

"use strict";

return g.isSupported();

};

function i(j, k) {

"use strict";

this.$WebWorkerPool0 = [];

for (var l = 0; l < k; l++)

this.$WebWorkerPool0.push(new g(this, l, j).setErrorHandler(this.$WebWorkerPool1).setMessageHandler(this.$WebWorkerPool2).execute());

this.$WebWorkerPool3 = 0;

}

i.prototype.setMessageHandler = function(j) {

"use strict";

this.$WebWorkerPool4 = j || h;

return this;

};

i.prototype.setAllowCrossPageTransition = function(j) {

"use strict";

this.$WebWorkerPool0.forEach(function(k) {

return k.setAllowCrossPageTransition(j);

});

return this;

};

i.prototype.setErrorHandler = function(j) {

"use strict";

this.$WebWorkerPool5 = j || h;

return this;

};

i.prototype.postMessage = function() {

"use strict";

for (var j = [], k = 0, l = arguments.length; k < l; k++)

j.push(arguments[k]);

var m = this.getWorker();

m.postMessage.apply(m, j);

return m;

};

i.prototype.$WebWorkerPool2 = function() {

"use strict";

for (var j = [], k = 0, l = arguments.length; k < l; k++)

j.push(arguments[k]);

j.unshift(this);

this.getPool().$WebWorkerPool4.apply(this.getPool(), j);

};

i.prototype.$WebWorkerPool1 = function() {

"use strict";

for (var j = [], k = 0, l = arguments.length; k < l; k++)

j.push(arguments[k]);

j.unshift(this);

return this.getPool().$WebWorkerPool5.apply(this.getPool(), j);

};

i.prototype.getWorker = function() {

"use strict";

var j = this.$WebWorkerPool0[this.$WebWorkerPool3];

this.$WebWorkerPool3 = (this.$WebWorkerPool3 + 1)%this.$WebWorkerPool0.length;

return j;

};

e.exports = i;

}, null);

\_\_d("JpegResizer", ["Emscripten", "JpegResizerConfig", "JpegResizeWorkerResource", "WebWorker", "WebWorkerPool", "PUWErrorCodes"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = null;

n.isSupported = function() {

"use strict";

return k.isSupported() && g.isSupported()&&!h.isGKBlacklisted;

};

n.getSingleton = function() {

"use strict";

if (!m)

m = new n(n.DEFAULT\_POOL\_SIZE).setAllowCrossPageTransition(true);

return m;

};

n.prepareResource = function(o) {

"use strict";

j.prepareResource(i, o);

};

function n(o) {

"use strict";

o = o || n.DEFAULT\_POOL\_SIZE;

j.prepareResource(i);

this.$JpegResizer0 = (new k(i, o)).setErrorHandler(this.$JpegResizer1.bind(this)).setMessageHandler(this.$JpegResizer2.bind(this));

this.$JpegResizer3 = {};

this.$JpegResizer4 = n.DEFAULT\_RESIZE\_HEIGHT;

this.$JpegResizer5 = n.DEFAULT\_RESIZE\_WIDTH;

}

n.prototype.setHighQuality = function(o) {

"use strict";

if (o) {

this.$JpegResizer4 = n.DEFAULT\_RESIZE\_HEIGHT;

this.$JpegResizer5 = n.DEFAULT\_RESIZE\_WIDTH;

} else {

this.$JpegResizer4 = n.LOW\_QUALITY\_RESIZE\_HEIGHT;

this.$JpegResizer5 = n.LOW\_QUALITY\_RESIZE\_WIDTH;

}

return this;

};

n.prototype.setAllowCrossPageTransition = function(o) {

"use strict";

this.$JpegResizer0.setAllowCrossPageTransition(o);

return this;

};

n.prototype.resizeBlob = function(o, p, q) {

"use strict";

var r = this.$JpegResizer0.getWorker();

if (r.isCurrentState('terminated')) {

var s = new Error('Worker terminated');

s.code = l.WORKER\_TERMINATED;

p(s);

return this;

}

var t = r.postMessage({

blob: o,

height: this.$JpegResizer4,

width: this.$JpegResizer5

}).getIndex();

this.$JpegResizer6(t).unshift({

done: p,

progress: q

});

return this;

};

n.prototype.$JpegResizer6 = function(o) {

"use strict";

if (!this.$JpegResizer3.hasOwnProperty(o))

this.$JpegResizer3[o] = [];

return this.$JpegResizer3[o];

};

n.prototype.$JpegResizer1 = function(o, p) {

"use strict";

o.terminate();

var q = this.$JpegResizer6(o.getIndex());

this.$JpegResizer7(p);

q.forEach(function(r) {

return r.done(p);

});

q.length = 0;

};

n.prototype.$JpegResizer2 = function(o, p) {

"use strict";

if (p.error)

this.$JpegResizer7(p.error);

if (p.done) {

this.$JpegResizer6(o.getIndex()).pop().done(p.error, p.blob, p.skipped);

} else {

var q = this.$JpegResizer6(o.getIndex()), r = q[q.length - 1].progress;

if (r)

r({

written: p.written,

total: p.total

});

}

};

n.prototype.$JpegResizer7 = function(o) {

"use strict";

var p = o.message;

if (p.indexOf('Not a JPEG file')!==-1) {

o.code = l.BAD\_JPEG;

} else if (p.indexOf('abort()')!==-1) {

o.code = l.WORKER\_ABORT;

} else

o.code = l.UNKNOWN\_RESIZE\_ERROR;

};

n.DEFAULT\_POOL\_SIZE = 3;

n.DEFAULT\_RESIZE\_HEIGHT = 2048;

n.DEFAULT\_RESIZE\_WIDTH = 2048;

n.LOW\_QUALITY\_RESIZE\_HEIGHT = 960;

n.LOW\_QUALITY\_RESIZE\_WIDTH = 960;

e.exports = n;

}, null);

\_\_d("MercuryMessageActions", ["MercuryActionTypeConstants", "MercuryPayloadSource", "MercuryServerRequests", "MercurySingletonMixin"], function(a, b, c, d, e, f, g, h, i, j) {

'use strict';

function k(l) {

this.$MercuryMessageActions0 = l;

this.$MercuryMessageActions1 = i.getForFBID(this.$MercuryMessageActions0);

}

k.prototype.markSpam = function(l, m) {

this.$MercuryMessageActions1.handleUpdate(this.$MercuryMessageActions2(l, m, g.MARK\_MESSAGES\_SPAM, h.CLIENT\_MARK\_MESSAGES\_SPAM));

};

k.prototype["delete"] = function(l, m, n) {

this.$MercuryMessageActions1.handleUpdate(this.$MercuryMessageActions2(l, m, g.DELETE\_MESSAGES, n || h.CLIENT\_DELETE\_MESSAGES));

};

k.prototype.$MercuryMessageActions2 = function(l, m, n, o) {

return {

actions: [{

action\_type: n,

action\_id: null,

thread\_id: l,

message\_ids: m

}

],

from\_client: true,

payload\_source: o

};

};

Object.assign(k, j);

e.exports = k;

}, null);

\_\_d("MercuryFileUploader", ["ArbiterMixin", "CSS", "Dialog", "DOM", "Event", "FileForm", "FileFormResetOnSubmit", "FileInput", "FormSubmitOnChange", "MercuryAttachment", "MercuryAttachmentTemplates", "MercuryConstants", "PhotosUploadID", "SubscriptionsHandler", "csx", "fbt", "getObjectValues", "mixin", "shield", "startsWith", "tx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, aa) {

var ba = x(g);

for (var ca in ba)

if (ba.hasOwnProperty(ca))

ea[ca] = ba[ca];

var da = ba === null ? null: ba.prototype;

ea.prototype = Object.create(da);

ea.prototype.constructor = ea;

ea.\_\_superConstructor\_\_ = ba;

function ea(ga, ha, ia, ja) {

"use strict";

this.$MercuryFileUploader0 = ga;

this.$MercuryFileUploader1 = {};

this.$MercuryFileUploader2 = {};

this.$MercuryFileUploader3 = {};

this.$MercuryFileUploader4 = {};

this.$MercuryFileUploader5 = {};

this.$MercuryFileUploader6 = {};

this.$MercuryFileUploader7 = {};

this.$MercuryFileUploader8 = {};

this.updateElements(ha, ia, ja);

}

ea.prototype.updateElements = function(ga, ha, ia) {

"use strict";

this.$MercuryFileUploader9 && this.$MercuryFileUploader9.release();

this.$MercuryFileUploader9 = new t();

this.$MercuryFileUploadera && this.$MercuryFileUploadera.destroy();

this.$MercuryFileUploadera = new l(ga, [o, m]);

this.$MercuryFileUploadera.setAllowCrossOrigin(true);

this.$MercuryFileUploadera.setUploadInParallel(true);

var ja = j.find(ga, ".\_4q60"), ka = j.find(ja, ".\_4q61");

new n(ja, ka, ha);

this.$MercuryFileUploader9.addSubscriptions(this.$MercuryFileUploadera.subscribe('submit', function() {

var la = {

count: 0,

file\_sizes: []

};

if (ha.files) {

for (var ma = 0; ma < ha.files.length; ma++)

if (ha.files[ma].size > r.AttachmentMaxSize) {

this.showAttachmentSizeErrorDialog();

return false;

}

var na = {};

for (var oa = 0; oa < ha.files.length; oa++) {

var pa = this.$MercuryFileUploaderb();

this.$MercuryFileUploaderc(pa, ha.files[oa].name);

la.count++;

la.file\_sizes.push(ha.files[oa].size);

na[pa] = ha.files[oa];

}

this.$MercuryFileUploadera.setFiles(na);

} else {

ia.value = this.$MercuryFileUploaderb();

this.$MercuryFileUploaderc(ia.value, ha.value);

la.count = 1;

}

this.inform('submit', la);

}.bind(this)), this.$MercuryFileUploadera.subscribe('success', this.$MercuryFileUploaderd.bind(this)), this.$MercuryFileUploadera.subscribe('failure', this.$MercuryFileUploadere.bind(this)), k.listen(ka, 'click', y(this.inform, this, 'open')));

};

ea.prototype.showAttachmentSizeErrorDialog = function() {

"use strict";

this.$MercuryFileUploadera.abort();

this.$MercuryFileUploadera.clear();

new i().setTitle("Die von dir ausgew\u00e4hlte Datei ist zu gro\u00df").setBody("Die von dir ausgew\u00e4hlte Datei ist zu gro\u00df. Die maximale Gr\u00f6\u00dfe betr\u00e4gt 25 MB.").setButtons(i.OK).setSemiModal(true).show();

};

ea.prototype.addDroppedFiles = function(ga) {

"use strict";

if (!this.$MercuryFileUploadera ||!this.$MercuryFileUploadera.canUseXHR())

return;

var ha = {}, ia = {

count: 0,

file\_sizes: []

};

for (var ja = 0; ja < ga.length; ja++) {

var ka = ga[ja];

if (ka.size > r.AttachmentMaxSize) {

this.showAttachmentSizeErrorDialog();

return false;

}

var la = this.$MercuryFileUploaderb();

this.$MercuryFileUploaderc(la, ka.name);

ha[la] = ka;

ia.file\_sizes.push(ka.size);

ia.count++;

}

this.$MercuryFileUploadera.setFiles(ha);

this.$MercuryFileUploadera.forceSendViaXHR();

this.inform('submit', ia);

};

ea.prototype.isUploading = function() {

"use strict";

return !!Object.keys(this.$MercuryFileUploader6).length;

};

ea.prototype.addCachedAttachments = function(ga) {

"use strict";

ga.forEach(function(ha) {

var ia = this.$MercuryFileUploaderb();

this.$MercuryFileUploaderc(ia, ha.filename);

this.$MercuryFileUploaderf(ia, ha);

}.bind(this));

};

ea.prototype.addCachedImageFiles = function(ga) {

"use strict";

var ha = "Foto";

ga.forEach(function(ia) {

var ja = this.$MercuryFileUploaderb();

this.$MercuryFileUploaderc(ja, ha);

this.$MercuryFileUploaderf(ja, {

filename: ia,

image\_id: ia,

filetype: 'image/jpeg'

});

}.bind(this));

};

ea.prototype.getAttachments = function() {

"use strict";

return w(this.$MercuryFileUploader1);

};

ea.prototype.getImageFiles = function() {

"use strict";

var ga = Object.keys(this.$MercuryFileUploader2).sort(), ha = [];

ga.forEach(function(ia) {

return ha.push(this.$MercuryFileUploader2[ia]);

}.bind(this));

return ha;

};

ea.prototype.getVideoFiles = function() {

"use strict";

return w(this.$MercuryFileUploader3);

};

ea.prototype.getAudioFiles = function() {

"use strict";

return w(this.$MercuryFileUploader4);

};

ea.prototype.getFiles = function() {

"use strict";

return w(this.$MercuryFileUploader5);

};

ea.prototype.removeAttachments = function() {

"use strict";

j.empty(this.$MercuryFileUploader0);

this.$MercuryFileUploader1 = {};

this.$MercuryFileUploader2 = {};

this.$MercuryFileUploader3 = {};

this.$MercuryFileUploader4 = {};

this.$MercuryFileUploader5 = {};

this.$MercuryFileUploader7 = {};

this.$MercuryFileUploader6 = {};

this.$MercuryFileUploader8 = {};

h.hide(this.$MercuryFileUploader0);

this.inform('dom-updated');

};

ea.prototype.destroy = function() {

"use strict";

this.$MercuryFileUploader9 && this.$MercuryFileUploader9.release();

this.$MercuryFileUploadera && this.$MercuryFileUploadera.destroy();

this.removeAttachments();

};

ea.prototype.$MercuryFileUploaderc = function(ga, ha) {

"use strict";

var ia = q[':fb:mercury:upload-file-row'].build();

this.$MercuryFileUploader7[ga] = ia;

this.$MercuryFileUploader6[ga] = true;

this.$MercuryFileUploader8[ga] = Date.now();

j.appendContent(ia.getNode('iconText'), fa(ha));

k.listen(ia.getNode('closeFileUpload'), 'click', this.$MercuryFileUploaderg.bind(this, ga));

j.appendContent(this.$MercuryFileUploader0, ia.getRoot());

h.show(this.$MercuryFileUploader0);

this.inform('dom-updated');

};

ea.prototype.$MercuryFileUploaderg = function(ga, event) {

"use strict";

if (this.$MercuryFileUploader6[ga]) {

this.inform('upload-canceled-during-upload');

} else if (this.$MercuryFileUploader1[ga] || this.$MercuryFileUploader2[ga] || this.$MercuryFileUploader3[ga] || this.$MercuryFileUploader4[ga] || this.$MercuryFileUploader5[ga])

this.inform('upload-canceled-after-uploaded');

delete this.$MercuryFileUploader1[ga];

delete this.$MercuryFileUploader2[ga];

delete this.$MercuryFileUploader3[ga];

delete this.$MercuryFileUploader4[ga];

delete this.$MercuryFileUploader5[ga];

delete this.$MercuryFileUploader6[ga];

delete this.$MercuryFileUploader8[ga];

var ha = this.$MercuryFileUploader7[ga];

delete this.$MercuryFileUploader7[ga];

if (ha) {

j.remove(ha.getRoot());

this.inform('dom-updated');

}

this.inform('upload-canceled');

return false;

};

ea.prototype.$MercuryFileUploaderh = function(ga, ha) {

"use strict";

var ia = this.$MercuryFileUploader7[ga], ja = p.getAttachIconClassByMime(ha);

h.addClass(ia.getNode('iconText'), ja);

h.addClass(ia.getRoot(), 'done');

};

ea.prototype.$MercuryFileUploaderf = function(ga, ha) {

"use strict";

if (this.$MercuryFileUploader6[ga]) {

delete this.$MercuryFileUploader6[ga];

if (ha.image\_id) {

this.$MercuryFileUploader2[ga] = ha.image\_id;

} else if (ha.video\_id) {

this.$MercuryFileUploader3[ga] = ha.video\_id;

} else if (ha.audio\_id) {

this.$MercuryFileUploader4[ga] = ha.audio\_id;

} else if (ha.file\_id) {

this.$MercuryFileUploader5[ga] = ha.file\_id;

} else

this.$MercuryFileUploader1[ga] = ha;

this.$MercuryFileUploaderh(ga, ha.filetype);

this.inform('one-upload-completed', {

upload\_time\_ms: Date.now() - this.$MercuryFileUploader8[ga]

});

}

if (!this.isUploading())

this.inform('all-uploads-completed', {

count: this.getAttachments().length

});

};

ea.prototype.$MercuryFileUploaderd = function(event, ga) {

"use strict";

var ha = ga.response.getPayload();

this.$MercuryFileUploaderf(this.$MercuryFileUploaderi(ga), ha.metadata[0]);

};

ea.prototype.$MercuryFileUploadere = function(event, ga) {

"use strict";

this.inform('one-upload-failed');

this.$MercuryFileUploaderg(this.$MercuryFileUploaderi(ga), event);

};

ea.prototype.$MercuryFileUploaderb = function() {

"use strict";

return 'upload\_' + s.getNewID();

};

ea.prototype.$MercuryFileUploaderi = function(ga) {

"use strict";

var ha = ga.response.getPayload();

if (ga.upload) {

return ga.upload.getName();

} else

return ha.uploadID;

};

function fa(ga) {

if (ga && z(ga, 'C:\\fakepath\\'))

return ga.substring(12);

return ga;

}

e.exports = ea;

}, null);

\_\_d("MercuryShareLinkUploader", ["ArbiterMixin", "AsyncRequest", "CSS", "DOM", "Event", "Form", "URLScraper", "WebMessengerEvents", "mixin", "isEmpty"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

var q = o(g);

for (var r in q)

if (q.hasOwnProperty(r))

t[r] = q[r];

var s = q === null ? null: q.prototype;

t.prototype = Object.create(s);

t.prototype.constructor = t;

t.\_\_superConstructor\_\_ = q;

function t(u, v, w, x, y) {

"use strict";

this.$MercuryShareLinkUploader0 = u;

this.$MercuryShareLinkUploader1 = v;

this.$MercuryShareLinkUploader2 = w;

this.$MercuryShareLinkUploader3 = y;

this.$MercuryShareLinkUploader4 = false;

this.$MercuryShareLinkUploader5 = null;

this.$MercuryShareLinkUploader6 = null;

k.listen(x, 'click', this.close.bind(this));

k.listen(y, 'keyup', function() {

!y.value.length && this.$MercuryShareLinkUploader7.enable();

}.bind(this));

this.$MercuryShareLinkUploader7 = new m(y);

this.$MercuryShareLinkUploader7.subscribe('match', function(z, aa) {

this.loadShare(aa && aa.url);

}.bind(this));

}

t.prototype.getAttachData = function() {

"use strict";

return this.loadAttachData(this.$MercuryShareLinkUploader4, this.$MercuryShareLinkUploader0, this.$MercuryShareLinkUploader3);

};

t.prototype.loadAttachData = function(u, v, w, x) {

"use strict";

if (u) {

var y = l.serialize(v), z = y.attachment && y.attachment.params, aa = y.link\_metrics ? y.link\_metrics.no\_image: true;

if (!z || p(z))

return null;

var ba = x ? x(): w.value;

if (ba.indexOf(z.url)===-1)

return y;

if (!z.summary&&!z.favicon && aa && z.title && z.url && z.url.substr(0, z.title.length) === z.title)

return null;

return y;

}

return null;

};

t.prototype.check = function() {

"use strict";

this.$MercuryShareLinkUploader7.check();

};

t.prototype.close = function() {

"use strict";

this.$MercuryShareLinkUploader8();

this.$MercuryShareLinkUploader7.disable();

this.inform('closed');

};

t.prototype.clear = function() {

"use strict";

this.$MercuryShareLinkUploader8();

this.$MercuryShareLinkUploader7.enable();

};

t.prototype.enable = function() {

"use strict";

this.$MercuryShareLinkUploader7.enable();

};

t.prototype.disable = function() {

"use strict";

this.$MercuryShareLinkUploader7.disable();

};

t.prototype.loadShare = function(u) {

"use strict";

this.$MercuryShareLinkUploader5 = u;

this.$MercuryShareLinkUploader6 && this.$MercuryShareLinkUploader6.abort();

this.$MercuryShareLinkUploader6 = new h().setMethod('POST').setURI('/ajax/share\_scrape.php').setData({

u: u

}).setHandler(this.$MercuryShareLinkUploader9.bind(this).bind(null, u)).setStatusElement(this.$MercuryShareLinkUploader2);

this.$MercuryShareLinkUploader6.send();

this.inform('link-detected');

};

t.prototype.$MercuryShareLinkUploader9 = function(u, v) {

"use strict";

if (this.$MercuryShareLinkUploader5 !== u)

return;

this.$MercuryShareLinkUploader5 = null;

this.$MercuryShareLinkUploader6 = null;

i.show(this.$MercuryShareLinkUploader0);

j.empty(this.$MercuryShareLinkUploader1);

j.setContent(this.$MercuryShareLinkUploader1, v.payload);

this.$MercuryShareLinkUploader4 = true;

if (!this.getAttachData()) {

this.close();

return;

}

n.detailDOMChanged();

};

t.prototype.$MercuryShareLinkUploader8 = function() {

"use strict";

i.hide(this.$MercuryShareLinkUploader0);

this.$MercuryShareLinkUploader4 = false;

this.$MercuryShareLinkUploader5 = null;

this.$MercuryShareLinkUploader6 && this.$MercuryShareLinkUploader6.abort();

this.$MercuryShareLinkUploader6 = null;

this.$MercuryShareLinkUploader7.reset();

n.detailDOMChanged();

};

e.exports = t;

}, null);

\_\_d("XSkypeDeprecationWarningDialogControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/videocall\/skype\_deprecation\_warning\/", {

\_\_asyncDialog: {

type: "Int"

}

});

}, null);

\_\_d("VideoCallSkypeDeprecationWarning", ["AsyncDialog", "AsyncRequest", "CacheStorage", "VideoCallSupport", "XSkypeDeprecationWarningDialogControllerURIBuilder", "FBRTCLogger"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = 'localstorage', n = 'RTC\_', o = 'SKYPE\_DEPRECATION\_STATE', p = 1000 \* 60 \* 60 \* 24;

function q(r) {

"use strict";

this.$VideoCallSkypeDeprecationWarning0 = new i(m, n);

this.$VideoCallSkypeDeprecationWarning1 = r;

this.$VideoCallSkypeDeprecationWarning2 = l.getInstance();

}

q.prototype.showWarningOrStartCall = function(r, s) {

"use strict";

if (this.$VideoCallSkypeDeprecationWarning3(r)) {

this.$VideoCallSkypeDeprecationWarning4(r);

} else if (this.$VideoCallSkypeDeprecationWarning5(r)) {

this.$VideoCallSkypeDeprecationWarning1.makeWebRTCCall(r, s || l.Trigger.CHAT\_TAB\_ICON);

} else

this.$VideoCallSkypeDeprecationWarning1.makeSkypeCall(r);

};

q.prototype.$VideoCallSkypeDeprecationWarning5 = function(r) {

"use strict";

if (!this.$VideoCallSkypeDeprecationWarning1.canCallByWebrtc(r))

return false;

if (!j.isPluginInstalled(false))

return true;

return this.$VideoCallSkypeDeprecationWarning6().optedInToWebRTC;

};

q.prototype.$VideoCallSkypeDeprecationWarning3 = function(r) {

"use strict";

if (!this.$VideoCallSkypeDeprecationWarning1.canCallByWebrtc(r))

return false;

if (!j.isPluginInstalled(false))

return false;

var s = this.$VideoCallSkypeDeprecationWarning6();

return !s.optedInToWebRTC && (Date.now() - s.updatedAt) > p;

};

q.prototype.$VideoCallSkypeDeprecationWarning4 = function(r) {

"use strict";

this.$VideoCallSkypeDeprecationWarning2.logInfo(r, null, 'Show Deprecation Dialog');

var s = new k().getURI(), t = new h(s), u = this;

this.$VideoCallSkypeDeprecationWarning7();

g.send(t, function(v) {

v.subscribe('confirm', function() {

v.hide();

u.$VideoCallSkypeDeprecationWarning8();

u.$VideoCallSkypeDeprecationWarning1.makeWebRTCCall(r, l.Trigger.SKYPE\_DEPRECATION\_DIALOG);

u.$VideoCallSkypeDeprecationWarning2.logCallAction(r, null, l.CallAction.TRY\_NEW);

});

v.subscribe('cancel', function() {

u.$VideoCallSkypeDeprecationWarning1.makeSkypeCall(r);

u.$VideoCallSkypeDeprecationWarning2.logCallAction(r, null, l.CallAction.START\_SKYPE);

});

});

};

q.prototype.$VideoCallSkypeDeprecationWarning7 = function() {

"use strict";

this.$VideoCallSkypeDeprecationWarning0.set(o, {

optedInToWebRTC: false,

updatedAt: Date.now()

});

};

q.prototype.$VideoCallSkypeDeprecationWarning8 = function() {

"use strict";

this.$VideoCallSkypeDeprecationWarning0.set(o, {

optedInToWebRTC: true,

updatedAt: Date.now()

});

};

q.prototype.$VideoCallSkypeDeprecationWarning6 = function() {

"use strict";

var r = this.$VideoCallSkypeDeprecationWarning0.get(o);

if (r)

return r;

return {

optedInToWebRTC: false,

updatedAt: 0

};

};

q.prototype.ut\_setStorage = function(r) {

"use strict";

this.$VideoCallSkypeDeprecationWarning0 = r;

};

e.exports = q;

}, null);

\_\_d("VideoCallRecordMessageDialog", ["AsyncDialog", "AsyncRequest", "Dialog", "URI", "tx"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = {

get: function(m, n) {

var o = "M\u00f6chtest du eine Nachricht hinterlassen?", p = "Neue Nachricht";

return new i().setTitle(k.\_("{firstname} ist nicht verf\u00fcgbar", {

firstname: n

})).setBody(o).setButtons([{

name: 'record-message',

label: p

}, i.CANCEL]).setHandler(function() {

var q = j('/ajax/messaging/composer.php').setQueryData({

ids: [m],

autoloadvideo: true

}).toString();

g.send(new h(q));

});

}

};

e.exports = l;

}, null);

\_\_d("VideoCallCore", ["Event", "Arbiter", "AsyncRequest", "AvailableListConstants", "Bootloader", "ChannelConstants", "Cookie", "CSS", "Dialog", "UserAgent\_DEPRECATED", "VideoCallSupport", "VideoCallSkypeDeprecationWarning", "emptyFunction", "ge", "PresenceStatus", "randomInt", "VideoCallUI", "VideoCallIncomingCallController", "VideoCallTemplates", "ShortProfiles", "VideoCallRecordMessageDialog"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v) {

var w = b('VideoCallUI').module, x = b('VideoCallIncomingCallController').module;

b('VideoCallTemplates');

var y = [], z = [], aa = null, ba = {

mightReloadPostInstall: function() {

return p.windows();

},

onVideoMessage: function(ga) {

y.push(ga);

k.loadModules(["VideoCallController"], s);

},

onRTCMessage: function(ga) {

if (q.isReceiveWebrtcSupported() && x) {

z.push(ga);

ca();

}

},

setMessageHandler: function(ga) {

this.onVideoMessage = ga;

if (ga)

while (y.length)

ga(y.shift());

},

setRTCMessageHandler: function(ga) {

this.onRTCMessage = ga;

if (ga)

while (z.length)

ga(z.shift());

},

availableForCall: function(ga) {

var ha = u.get(ga);

if (ha === j.ACTIVE) {

var ia = u.getDetailedActivePresence(ga);

if (ia === j.ACTIVE\_ON\_WEB)

return true;

}

return ea(ga);

},

attachListenerToProfileButton: function(ga) {

var ha = t('videoCallProfileButton');

if (ha) {

if (!q.isVideoCallSupported()) {

n.hide(ha);

return;

}

g.listen(ha, 'click', function(event) {

ba.startCallOrLeaveMessage(ga, 'profile\_button\_click\_timeline');

});

}

},

startCallOrLeaveMessage: function(ga, ha) {

if (this.availableForCall(ga)) {

ba.showOutgoingCallDialog(ga, ha);

} else

b('ShortProfiles').get(ga, function(ia) {

b('VideoCallRecordMessageDialog').get(ga, ia.firstName).show();

});

},

showOutgoingCallDialog: function(ga, ha, ia) {

ba.logClick(ga, (ha || 'unknown'));

var ja = new r(ba);

ja.showWarningOrStartCall(ga, ia);

},

canCallByWebrtc: function(ga) {

if (q.isSendWebrtcSupported() && w)

return ea(ga);

return false;

},

makeWebRTCCall: function(ga, ha) {

var ia = parseInt(ga, 10);

ca();

var ja = da();

aa.startingCallTo(ia, ja);

w.openAsCaller(ga, ja, ha);

},

makeSkypeCall: function(ga) {

var ha = q.isPluginInstalled() ? 'outgoing\_dialog.php': 'intro.php', ia = '/ajax/chat/video/' + ha + '?idTarget=' + ga;

new o().setAllowCrossPageTransition(true).setAsync(new i(ia).setHandler(s).setServerDialogCancelHandler(s)).show();

},

logClick: function(ga, ha) {

new i().setURI('/ajax/chat/video/log\_click.php').setData({

targetUserID: ga,

clickSource: ha

}).setAllowCrossPageTransition(true).setErrorHandler(s).send();

}

};

function ca() {

if (!aa)

aa = new x(ba);

}

function da() {

return v(0, 4294967295);

}

function ea(ga) {

var ha = u.getCapabilities(ga), ia = l.CAPABILITY\_VOIP\_INTEROP | l.CAPABILITY\_VIDEO, ja = ha & ia;

return ja !== 0;

}

function fa() {

if (!ba.mightReloadPostInstall())

return;

var ga = m.get('vcpwn');

if (ga) {

m.clear('vcpwn');

var ha = m.get('vctid');

if (ha) {

m.clear('vctid');

if (m.get('vctid'))

return;

if (ha && ba.isInstalled()) {

var ia = '/ajax/chat/video/outgoing\_dialog.php?idTarget=' + ha;

new o().setAllowCrossPageTransition(true).setAsync(new i(ia)).show();

}

}

}

}

h.subscribe(l.getArbiterType('video'), function(ga, ha) {

ba.onVideoMessage(ha.obj);

});

h.subscribe(l.getArbiterType('webrtc'), function(ga, ha) {

ba.onRTCMessage(ha.obj);

});

h.subscribe(l.getArbiterType('chat\_event'), function(ga, ha) {

if (ha.obj.event\_name == "missed-call")

k.loadModules(["VideoCallController"], function(ia) {

ia.onMissedCallEvent(ha.obj);

});

});

fa();

e.exports = ba;

}, null);

\_\_d("ChatActivity", ["Event", "Arbiter", "AvailableList", "AvailableListConstants", "JSLogger", "MercuryConfig", "PresenceState", "UserActivity", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o) {

var p = l.activity\_limit || 60000, q = l.idle\_limit || 1800000, r = l.idle\_poll\_interval || 300000, s = k.create('chat\_activity'), t = Date.now(), u = t, v = true;

function w() {

var aa = Date.now();

return !!(v && (aa - t < p));

}

var x = o(new h(), {

isActive: w

});

function y() {

var aa = t;

t = Date.now();

if (t - aa > q) {

s.debug('idle\_to\_active', aa);

m.doSync();

}

x.inform('activity');

}

i.subscribe(j.ON\_AVAILABILITY\_CHANGED, function() {

if (!i.isUserIdle())

u = Date.now();

});

g.listen(window, 'focus', function() {

v = true;

y();

});

g.listen(window, 'blur', function() {

v = false;

});

n.subscribe(function() {

y();

});

function z(aa) {

var ba = aa && aa.at && m.verifyNumber(aa.at);

if (typeof ba !== 'number')

ba = null;

return ba || 0;

}

setInterval(function() {

var aa = Date.now(), ba = z(m.get()), ca = Math.max(t, ba, u);

if (aa - ca > q) {

s.debug('idle', {

cookie: ba,

local: t,

presence: u

});

x.inform('idle', aa - ca);

}

}, r);

m.registerStateStorer(function(aa) {

var ba = z(aa);

if (ba < t)

aa.at = t;

return aa;

});

h.subscribe(k.DUMP\_EVENT, function(aa, ba) {

ba.chat\_activity = {

activity\_limit: p,

idle\_limit: q,

idle\_poll\_interval: r,

last\_active\_time: t,

last\_global\_active\_time: u

};

});

e.exports = x;

}, null);

\_\_d("ChatAnimatedGifs", ["URI"], function(a, b, c, d, e, f, g) {

var h = {

shouldHideBody: function(i) {

if (!i.has\_attachment)

return false;

for (var j = 0; j < i.attachments.length; j++) {

var k = i.attachments[j];

if (k.preview\_url) {

var l = h.getRawUrlFromSafeUrl(k.preview\_url);

if (i.body == l)

return true;

}

}

return false;

},

getRawUrlFromSafeUrl: function(i) {

return g(i).getQueryData().url;

}

};

e.exports = h;

}, null);

\_\_d("MercuryTimestampTracker", ["MercuryActionTypeConstants", "MercuryPayloadSource", "MercurySingletonMixin", "MercuryServerRequests", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k) {

function l(m) {

this.\_fbid = m;

this.\_serverRequests = j.getForFBID(this.\_fbid);

this.\_lastTimestamp = 0;

this.\_serverRequests.subscribe('update-messages', function(n, o) {

if (!o.actions ||!o.actions.length)

return;

if (o.payload\_source == h.CLIENT\_SEND\_MESSAGE || o.payload\_source == h.UNKNOWN)

return;

for (var p = 0; p < o.actions.length; p++) {

var q = o.actions[p], r = q.action\_type;

if (r == g.USER\_GENERATED\_MESSAGE && q.thread\_id && q.timestamp > this.\_lastTimestamp)

this.\_lastTimestamp = q.timestamp;

}

}.bind(this));

}

k(l.prototype, {

getLastUserMessageTimestamp: function() {

return this.\_lastTimestamp;

}

});

k(l, i);

e.exports = l;

}, null);

\_\_d("ChatTabModel", ["Arbiter", "ArbiterMixin", "ChatBehavior", "ChatConfig", "LogHistory", "JSLogger", "MercuryAssert", "MercuryLocalIDs", "PresenceInitialData", "PresenceState", "PresenceUtil", "areJSONRepresentationsEqual", "arrayContains", "copyProperties", "MercuryServerRequests", "MercuryThreads", "MercuryTimestampTracker"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t) {

var u = b('MercuryServerRequests').get(), v = b('MercuryThreads').get(), w = b('MercuryTimestampTracker').get(), x = [], y = null, z = null, aa = null;

function ba() {

return parseInt(o.serverTime, 10);

}

var ca = j.get('tab\_max\_load\_age') || 3600000, da = ba() - ca, ea = 0, fa = 20, ga = k.getInstance('chat\_tab\_model'), ha = false;

function ia(bb) {

var cb = p.verifyNumber(bb.uct2);

if (!cb || typeof cb !== 'number') {

ga.warn('bad\_cookie\_version', bb);

return null;

}

if (cb < ea || cb < da)

return null;

return cb;

}

function ja(bb) {

if (bb.uct2 == ea)

return;

var cb = ia(bb), db = {};

if (!cb) {

db.old\_tabs = bb && bb.t2 && JSON.stringify(bb.t2);

db.old\_promoted = bb && bb.lm2;

db.old\_time = bb && bb.uct2;

db.old\_reason = bb && bb.tr;

db.old\_window = bb && bb.tw;

var eb = [];

x.forEach(function(fb) {

if (!fb.fragile) {

var gb = {

i: fb.id,

si: fb.server\_id

};

if (fb.raised)

gb.r = 1;

eb.push(gb);

}

});

bb.t2 = eb;

bb.lm2 = y;

bb.uct2 = ea;

bb.tr = aa;

bb.tw = q.getSessionID();

db.new\_tabs = JSON.stringify(bb.t2);

db.new\_promoted = bb.lm2;

db.new\_time = bb.uct2;

db.new\_reason = bb.tr;

db.new\_window = bb.tw;

ga.debug('store', db);

} else {

db.tabs = bb && bb.t2 && JSON.stringify(bb.t2);

db.promoted = bb && bb.lm2;

db.time = bb && bb.uct2;

db.reason = bb && bb.tr;

db.window = bb && bb.tw;

db.last\_change\_time = ea;

db.last\_change\_reason = aa;

db.min\_change\_time = da;

ga.warn('store\_bad\_state', db);

}

return bb;

}

function ka(bb) {

if (bb) {

var cb = ia(bb);

if (cb && cb !== ea) {

var db = {};

db.old\_tabs = JSON.stringify(x);

db.old\_promoted = y;

db.old\_time = ea;

db.old\_reason = aa;

db.window\_id = q.getSessionID();

db.cookie\_tabs = bb && bb.t2 && JSON.stringify(bb.t2);

db.cookie\_promoted = bb && bb.lm2;

db.cookie\_time = bb && bb.uct2;

db.cookie\_reason = bb && bb.tr;

db.cookie\_window = bb && bb.tw;

ea = cb;

aa = 'load';

var eb = la(bb.t2, bb.lm2 || null);

db.load\_result = eb;

db.new\_tabs = JSON.stringify(x);

db.new\_promoted = y;

db.new\_time = ea;

db.new\_reason = aa;

var event = 'load';

if (!ha)

event += '\_init';

ga.log(event, db);

return eb;

}

} else

ga.warn('load\_bad\_state', bb);

return false;

}

function la(bb, cb) {

if (ma(bb, cb)) {

var db = x.filter(function(gb) {

return gb.fragile;

}), eb = {};

y = null;

x = bb.map(function(gb) {

var hb = {

id: gb.i,

server\_id: gb.si

};

if (hb.id == cb)

y = hb.id;

if (gb.r)

hb.raised = true;

eb[hb.id] = hb;

return hb;

});

x = x.filter(function(gb) {

return gb != null;

});

if (z)

for (var fb in z)

if (!eb[fb] ||!eb[fb].raised)

delete z[fb];

db = db.filter(function(gb) {

return !eb[gb.id];

});

x = x.concat(db);

qa();

return true;

}

return false;

}

function ma(bb, cb) {

if (cb != y)

return true;

var db = x.filter(function(gb) {

return !gb.fragile;

});

if (bb.length != db.length)

return true;

for (var eb = 0, fb = bb.length; eb < fb; eb++)

if (!r(bb[eb], db[eb]))

return true;

return false;

}

function na(bb, cb, db) {

var eb = ka(p.get());

if (cb === undefined || cb > ea) {

if (bb()) {

eb = true;

aa = db || null;

pa(cb);

}

} else

ga.error('rejected', {

change\_time: cb,

state\_time: ea

});

eb && oa();

}

function oa() {

if (ha)

za.inform('chat/tabs-changed', za.get());

}

function pa(bb) {

if (bb === undefined)

bb = Math.max(w.getLastUserMessageTimestamp() || 1, ea + 1);

ea = bb;

p.doSync();

}

function qa(bb) {

var cb = x.length - fa;

if (cb > 0)

x = x.filter(function(db) {

return db.raised || db.id == bb || cb--<=0;

});

if (cb > 0)

x = x.filter(function(db) {

return db.id == y || db.id == bb || cb--<=0;

});

}

function ra(bb) {

for (var cb = 0; cb < x.length; cb++)

if (x[cb].id == bb)

return cb;

return - 1;

}

function sa(bb, cb) {

var db = v.getThreadMetaNow(bb);

if (!db)

return false;

if (db.is\_canonical\_user) {

return va(bb, cb);

} else {

var eb = ta(bb);

if (eb)

u.getServerThreadID(bb, function(fb) {

if (xa(bb, fb)) {

pa();

oa();

}

});

return eb;

}

}

function ta(bb) {

if (ra(bb)===-1) {

x.push({

id: bb,

fragile: true

});

ga.log('open\_fragile\_tab', {

tabs: JSON.stringify(x),

opened: bb,

window\_id: q.getSessionID()

});

return true;

}

return false;

}

function ua(bb) {

var cb = [];

bb.forEach(function(db) {

var eb = ra(db);

if (eb!==-1)

cb.push(x[eb]);

});

x.forEach(function(db) {

if (!s(bb, db.id))

cb.push(db);

});

na(function() {

if (ma(cb, y)) {

x = cb;

return true;

}

return false;

});

}

function va(bb, cb) {

var db = ra(bb);

if (db!=-1)

if (x[db].fragile) {

x.splice(db, 1);

} else {

cb && (x[db].signatureID = cb);

return true;

}

for (var eb = 0; eb <= x.length; eb++)

if (eb == x.length || x[eb].fragile) {

x.splice(eb, 0, {

id: bb,

signatureID: cb

});

qa(bb);

ga.log('open\_tab', {

tabs: JSON.stringify(x),

opened: bb,

window\_id: q.getSessionID()

});

return true;

}

}

function wa(bb) {

var cb = ra(bb);

if (cb!=-1 && (!x[cb].raised || y !== bb)) {

x[cb].raised = true;

y = bb;

return true;

}

return false;

}

function xa(bb, cb) {

var db = ra(bb);

if (db!=-1 && x[db].fragile) {

var eb = x[db];

eb.fragile = false;

eb.server\_id = cb;

var fb = [];

x.forEach(function(gb) {

if (gb.id != bb) {

if (eb && gb.fragile) {

fb.push(eb);

eb = null;

}

fb.push(gb);

}

});

if (eb)

fb.push(eb);

x = fb;

ga.log('make\_permanent', {

tabs: JSON.stringify(x),

tab\_id: bb,

window\_id: q.getSessionID()

});

return true;

}

return false;

}

function ya(bb) {

var cb = ra(bb);

if (bb == y)

y = null;

if (cb!=-1) {

x.splice(cb, 1);

ga.log('close\_tab', {

tabs: JSON.stringify(x),

closed: bb,

window\_id: q.getSessionID()

});

return true;

}

return false;

}

p.registerStateStorer(ja);

p.registerStateLoader(function(bb) {

if (ka(bb))

oa();

});

function za() {}

t(za, h, {

isTabPromoted: function(bb) {

if (bb)

return bb == y;

return false;

},

indexOf: function(bb) {

return ra(bb);

},

getTab: function(bb) {

m.isThreadID(bb);

var cb = this.indexOf(bb);

if (cb>-1) {

var db = x[cb];

return t({}, db);

}

return null;

},

getEmptyTab: function() {

var bb;

for (var cb = 0; cb < x.length; cb++) {

bb = x[cb].id;

if (v.isNewEmptyLocalThread(bb))

return bb;

}

bb = n.generateThreadID();

v.createNewLocalThread(bb, []);

return bb;

},

getServerTime: function() {

return ba();

},

changeShowingTabs: function(bb) {

return ua(bb);

},

closeAllTabs: function() {

if (x.length) {

ga.log('close\_all\_tabs', {

closed\_tabs: JSON.stringify(x),

window\_id: q.getSessionID()

});

x = [];

y = null;

if (z)

z = {};

pa();

oa();

}

},

closeFragileTabs: function() {

var bb = [];

for (var cb = 0; cb < x.length; cb++)

if (x[cb].fragile&&!v.isNewEmptyLocalThread(x[cb].id)) {

bb.push(x[cb]);

x.splice(cb);

oa();

break;

}

ga.log('close\_fragile\_tabs', {

tabs: JSON.stringify(x),

fragile\_closed: bb,

window\_id: q.getSessionID()

});

},

closeTab: function(bb, cb) {

m.isThreadID(bb);

var db = false;

if (z) {

delete z[bb];

db = true;

}

na(function() {

if (ya(bb))

db = true;

return db;

}, undefined, cb);

},

raiseTab: function(bb, cb, db) {

m.isThreadID(bb);

var eb = false;

if (z && cb) {

z[bb] = true;

eb = true;

}

na(function() {

if (sa(bb, db))

eb = true;

var fb = ra(bb);

if (fb!=-1&&!x[fb].raised) {

x[fb].raised = true;

eb = true;

ga.log('raise\_tab', {

tabs: JSON.stringify(x),

raised: bb,

window\_id: q.getSessionID()

});

}

return eb;

});

},

get: function() {

var bb = x.map(function(cb) {

var db = t({}, cb);

delete db.fragile;

if (z)

db.raised = z[db.id];

return db;

});

return {

tabs: bb,

promoted: y

};

},

openFragileTab: function(bb) {

m.isThreadID(bb);

if (ta(bb))

oa();

},

openTab: function(bb) {

m.isThreadID(bb);

na(sa.bind(null, bb));

},

lowerTab: function(bb) {

m.isThreadID(bb);

var cb = false;

if (z) {

delete z[bb];

cb = true;

}

na(function() {

var db = ra(bb);

if (db!=-1 && x[db].raised) {

delete x[db].raised;

ga.log('lower\_tab', {

tabs: JSON.stringify(x),

lowered: bb,

window\_id: q.getSessionID()

});

cb = true;

}

return cb;

});

},

raiseAndPromoteTab: function(bb, cb, db, eb, fb) {

m.isThreadID(bb);

var gb = false;

if (z && cb) {

z[bb] = true;

gb = true;

}

na(function() {

if (sa(bb, db))

gb = true;

if (wa(bb)) {

gb = true;

ga.log('raise\_and\_promote\_tab', {

tabs: JSON.stringify(x),

promoted: bb,

window\_id: q.getSessionID()

});

}

return gb;

}, eb, fb);

},

setPromotedTab: function(bb) {

m.isThreadID(bb);

na(wa.bind(null, bb));

},

promoteThreadInPlaceOfThread: function(bb, cb, db, eb) {

m.isThreadID(bb);

m.isThreadID(cb);

na(function() {

var fb = ra(bb), gb = ra(cb);

if (eb) {

y = bb;

} else if (y === cb)

return true;

var hb = x[fb];

if (db) {

m.isThreadID(db);

var ib = ra(db), jb = x[gb];

x[gb] = hb;

x.splice(fb, 1);

x.splice(ib, 0, jb);

} else {

x[fb] = x[gb];

x[gb] = hb;

}

return true;

});

},

persistLocalRaised: function() {

if (z) {

na(function() {

var bb = false;

x.forEach(function(cb) {

if (cb.raised != z[cb.id]) {

bb = true;

if (z[cb.id]) {

cb.raised = true;

} else

delete cb.raised;

}

});

return bb;

});

ga.log('persist\_local\_raise', {

tabs: JSON.stringify(x),

window\_id: q.getSessionID()

});

}

}

});

g.subscribe(l.DUMP\_EVENT, function(bb, cb) {

cb.chat\_tabs = {

promoted: y,

tabs: x.map(function(db) {

return t({}, db);

}),

time: ea,

max\_load\_age: ca

};

});

function ab() {

var bb = i.ignoresRemoteTabRaise();

if (bb&&!z) {

ga.log('start\_ignore\_remote\_raise');

z = {};

oa();

} else if (!bb && z) {

ga.log('stop\_ignore\_remote\_raise');

z = null;

oa();

}

}

i.subscribe(i.ON\_CHANGED, ab);

ab();

ka(p.getInitial(), true);

if (ea === 0)

ea = ba() - 600000;

ha = true;

e.exports = za;

}, null);

\_\_d("TabsViewport", ["Arbiter", "ArbiterMixin", "ChatTabModel", "Dock", "DOM", "DOMDimensions", "Event", "Parent", "Vector", "ViewportBounds", "areJSONRepresentationsEqual", "copyProperties", "csx", "shield"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t) {

var u = 175, v = 269, w = 0;

function x() {

return u;

}

function y() {

return v;

}

function z(aa) {

this.\_root = aa;

var ba = this.\_recalculateWidth.bind(this);

m.listen(window, 'resize', ba);

j.subscribe('resize', ba);

g.subscribeOnce('sidebar/initialized', ba, g.SUBSCRIBE\_NEW);

i.subscribe('chat/tabs-changed', t(this.\_recalculateTabs, this, true));

this.\_recalculateWidth();

this.\_initialized = true;

}

r(z.prototype, h, {

\_root: null,

\_initialized: false,

\_availableWidth: 0,

\_maxShown: 1,

\_viewState: null,

\_recalculateWidth: function() {

var aa = z.\_getAvailableDockWidth(this.\_root), ba = Math.max(1, Math.floor(aa / y())), ca = ba != this.\_maxShown;

if (!this.\_viewState || ca || aa <= this.\_viewState.usedWidth || aa > this.\_viewState.widthToShowNext) {

this.\_availableWidth = aa;

this.\_maxShown = ba;

this.\_viewState = null;

this.\_recalculateTabs(ca);

}

},

\_onTabsChanged: function() {

if (this.\_initialized) {

this.inform('tabs-changed');

this.inform('max-to-show-changed', this.\_maxShown);

this.inform('max-to-show-change-completed');

}

},

\_recalculateTabs: function(aa) {

var ba = z.\_getTabsToShow(i.get(), this.\_availableWidth);

if (aa ||!q(this.\_viewState, ba)) {

this.\_viewState = ba;

this.\_onTabsChanged();

}

},

getMaxTabsToShow: function() {

return this.\_maxShown;

},

checkWidth: function() {

this.\_recalculateWidth();

},

hasRoomForRaisedTab: function() {

return this.\_availableWidth - this.\_viewState.usedWidth > y();

},

getTabsToShow: function() {

return JSON.parse(JSON.stringify(this.\_viewState.tabsToShow));

},

getShowingTabsOrder: function() {

var aa = i.get(), ba = this.\_viewState.tabsToShow;

return aa.tabs.filter(function(ca) {

return ba[ca.id];

});

},

getRightmostHiddenTab: function() {

var aa = i.get(), ba = this.\_viewState.tabsToShow, ca = aa.tabs.filter(function(ea) {

return !ba[ea.id];

}), da = ca.shift();

return da ? da.id : null;

},

getLeftmostVisibleTab: function() {

var aa = this.getShowingTabsOrder(), ba = aa.pop();

return ba ? ba.id : null;

},

getRightmostRaisedTab: function() {

var aa = this.getShowingTabsOrder(), ba = aa.length;

for (var ca = 0; ca < ba; ca++)

if (i.getTab(aa[ca].id).raised)

return aa[ca].id;

return null;

},

getRaisedVisibleTabs: function() {

var aa = [], ba = this.getShowingTabsOrder(), ca = ba.length;

for (var da = 0; da < ca; da++)

if (i.getTab(ba[da].id).raised)

aa.push(ba[da].id);

return aa;

},

shouldPromoteOnRaise: function(aa) {

if (!this.\_viewState.tabsToShow[aa])

return true;

if (this.\_viewState.nextToHide != aa)

return false;

var ba = i.getTab(aa), ca = ba && ba.raised;

return !ca && (this.\_availableWidth - this.\_viewState.usedWidth < y() - x());

}

});

r(z, {

\_getAvailableDockWidth: function(aa) {

var ba = l.getViewportWithoutScrollbarDimensions().width;

ba -= p.getLeft() + p.getRight();

ba -= 50;

var ca = n.byClass(aa, 'fbDock'), da = k.find(ca, ".\_56ox"), ea = o.getElementDimensions(da).x;

ba -= ea;

var fa = k.scry(ca, ".\_56oy");

ea += o.getElementDimensions(fa.pop()).x;

var ga = o.getElementDimensions(ca), ha = ga.x - ea;

if (ha > w)

w = ha;

ba -= w;

ba -= 15;

return Math.max(ba, 0);

},

\_getTabsToShow: function(aa, ba) {

ba = Math.max(ba, y() + 1);

function ca(sa, ta) {

return (sa.raised ||!ta) ? y() : x();

}

var da = JSON.parse(JSON.stringify(aa.tabs)), ea =- 1, fa = null;

if (aa.promoted)

da.forEach(function(sa, ta) {

if (sa.id === aa.promoted) {

ea = ta;

fa = sa;

}

});

var ga = 0, ha = false, ia = 0, ja=!fa;

da.forEach(function(sa, ta) {

var ua = ca(sa, ha);

ha = ha ||!sa.raised;

sa.leftmostOffset = ga + ua;

ga += ua;

if (sa.leftmostOffset < ba)

ia++;

ja|=ta == ea;

sa.alreadyPlacedPromoted = ja;

});

function ka(sa, ta, ua) {

var va = {};

for (var wa = 0; wa < ta; wa++) {

var xa = sa[wa];

if (!xa.alreadyPlacedPromoted && wa == ta - 1) {

va[ua] = true;

} else

va[xa.id] = true;

}

return va;

}

var la = ka(da, ia, aa.promoted), ma = ka(da, ia - 1, aa.promoted), na = null;

for (var oa in la)

if (!ma[oa])

na = oa;

var pa = da[ia - 1], qa = pa ? pa.leftmostOffset: 0, ra = Infinity;

if (ia < da.length)

ra = da[ia].leftmostOffset;

return {

nextToHide: na,

tabsToShow: la,

usedWidth: qa,

widthToShowNext: ra

};

}

});

e.exports = z;

}, null);

\_\_d("ChatTabPresence", ["AvailableList", "ChatTabModel", "MercuryIDs"], function(a, b, c, d, e, f, g, h, i) {

var j = {};

function k(l) {

var m = i.getUserIDFromThreadID(l);

if (m)

g.updateForID(m);

}

h.subscribe('chat/tabs-changed', function(event, l) {

l.tabs.forEach(function(m) {

if (m.raised&&!j[m.id])

k(m.id);

});

j = {};

l.tabs.forEach(function(m) {

if (m.raised)

j[m.id] = true;

});

});

e.exports = {};

}, null);

\_\_d("ChatMentionsNotifications", ["fbt", "CurrentUser", "MercuryParticipants"], function(a, b, c, d, e, f, g, h, i) {

var j = {

notifyIfMessageToMe: function(k) {

var l = 'fbid:' + h.getID(), m = k.body;

i.get(l, function(n) {

var o = new RegExp(n.short\_name, "i");

if (o.test(m))

j.\_notify(k);

});

},

\_notify: function(k) {

var l = 48, m = k.body;

if (m.length > l)

m = m.substr(0, l - 3) + "...";

new Notification("\"You were tagged in a message!\"", {

body: m,

icon: "http://facebook.com//images/icons-large/fb-xl.png"

});

}

};

e.exports = j;

}, null);

\_\_d("ChatAddToThreadButton.react", ["Link.react", "ReactComponentWithPureRenderMixin", "React", "TrackingNodes", "cx", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = i.createClass({

displayName: 'ChatAddToThreadButton',

mixins: [h],

propTypes: {

onClick: i.PropTypes.func,

shown: i.PropTypes.bool

},

render: function() {

var n = "Weitere Freunde zum Chat hinzuf\u00fcgen", o = j.getTrackingInfo(j.types.ADD\_TO\_THREAD);

return (i.createElement(g, {

'aria-label': n,

className: (("addToThread") + (' ' + "button") + (!this.props.shown ? ' ' + "hidden\_elem" : '')),

'data-ft': o,

'data-hover': "tooltip",

'data-tooltip-alignv': "top",

onClick: this.props.onClick

}));

}

});

e.exports = m;

}, null);

\_\_d("ChatArchiveWarning.react", ["Image.react", "ReactComponentWithPureRenderMixin", "React", "cx", "ix"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = i.createClass({

displayName: 'ChatArchiveWarning',

mixins: [h],

propTypes: {

isFBAtWork: i.PropTypes.bool,

shown: i.PropTypes.bool

},

render: function() {

var m = this.props.isFBAtWork ? k('images/chat/tab/archive-grey.png'): k('images/chat/tab/archive.png');

return (i.createElement(g, {

src: m,

className: (("titanArchiveWarning") + (' ' + "button") + (!this.props.shown ? ' ' + "hidden\_elem" : '')),

'data-hover': "tooltip",

'data-tooltip-alignh': "center",

'aria-label': "All employee-to-employee conversations will be archived"

}));

}

});

e.exports = l;

}, null);

\_\_d("ChatCloseButton.react", ["Link.react", "ReactComponentWithPureRenderMixin", "React", "TrackingNodes", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k) {

"use strict";

var l = i.createClass({

displayName: 'ChatCloseButton',

mixins: [h],

propTypes: {

onClick: i.PropTypes.func,

onKeyUp: i.PropTypes.func,

onMouseDown: i.PropTypes.func,

active: i.PropTypes.bool

},

getDefaultProps: function() {

return {

active: false

};

},

render: function() {

var m = j.getTrackingInfo(j.types.CLOSE\_BUTTON), n = "Bet\u00e4tige zum Schlie\u00dfen die ESC-Taste", o = "Reiter schlie\u00dfen";

return (i.createElement(g, {

'aria-label': this.props.active ? n: o,

className: "close button",

'data-ft': m,

'data-hover': "tooltip",

'data-tooltip-position': "above",

onClick: this.props.onClick,

onKeyUp: this.props.onKeyUp,

onMouseDown: this.props.onMouseDown

}));

}

});

e.exports = l;

}, null);

\_\_d("VideoCallTourDialog", ["ArbiterMixin", "LegacyContextualDialog", "CSS", "MercuryIDs", "ChatTabTemplates", "VideoCallCore", "VideoCallingTour", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n) {

function o() {

this.\_dialog = null;

}

n(o.prototype, g);

n(o.prototype, {

render: function(p, q) {

var r = j.getUserIDFromThreadID(q);

if (!r ||!l.availableForCall(r))

return;

var s = k[':fb:mercury:call:tour'].build();

this.\_dialog = new h();

this.\_dialog.init(s.getRoot()).setWidth(250).setAlignH('center').setContext(p).show();

i.addClass(this.\_dialog.getRoot(), 'uiContextualDialogWithFooterArrowBottom');

i.addClass(p, 'video\_call\_tour');

this.inform('chat/dialog-rendered', {

dialog: this,

thread\_id: q

});

m.inform('videocallingtour/end');

},

updatePosition: function() {

if (this.\_dialog && this.\_dialog.isShown())

this.\_dialog.updatePosition();

},

hide: function() {

if (this.\_dialog && this.\_dialog.isShown()) {

this.\_dialog.hide();

this.\_dialog = null;

}

}

});

e.exports = o;

}, null);

\_\_d("ChatContextualDialogController", ["Event", "ArbiterMixin", "ChatTabModel", "VideoCallingTour", "VideoCallTourDialog", "copyProperties", "setTimeoutAcrossTransitions"], function(a, b, c, d, e, f, g, h, i, j, k, l, m) {

var n = 60000, o = false, p = function(w, x) {

this.\_videoCallTour = new k();

this.\_threadID = w;

this.\_tabMainElement = x;

this.\_openDialog = null;

this.\_subscriptionTokens = [];

this.\_scrollListener = null;

this.\_timeout = null;

};

function q(w, event, x) {

if (w.\_openDialog)

w.\_openDialog.updatePosition();

}

function r(w) {

if (w.\_openDialog)

w.\_openDialog.updatePosition();

}

function s(w) {

if (w.\_openDialog) {

w.\_openDialog.hide();

w.\_openDialog = null;

}

if (w.\_timeout) {

clearTimeout(w.\_timeout);

w.\_timeout = null;

}

while (w.\_subscriptionTokens.length)

w.\_subscriptionTokens.pop().unsubscribe();

if (w.\_scrollListener) {

w.\_scrollListener.remove();

w.\_scrollListener = null;

}

}

function t(w, event, x) {

if (x.thread\_id === w.\_threadID) {

w.\_openDialog = x.dialog;

v(w);

w.\_timeout = m(w.destroy.bind(w, w.\_threadID), n);

w.\_scrollListener = g.listen(window, 'scroll', r.bind(null, w));

}

}

function u(w, x) {

if (!w.\_openDialog) {

w.\_subscriptionTokens.push(x.subscribe('chat/dialog-rendered', t.bind(null, w)));

x.render(w.\_tabMainElement, w.\_threadID);

}

}

function v(w) {

w.\_subscriptionTokens.push(i.subscribe('chat/tabs-changed', q.bind(null, w)), p.subscribe('dialog/close-all', s.bind(null, w)));

}

l(p, h);

l(p.prototype, {

destroy: function() {

s(this);

},

tabFocused: function() {

if (o) {

u(this, this.\_videoCallTour);

return;

}

},

tabNotActive: function() {

s(this);

},

hideVideoCallouts: function() {

s(this);

}

});

j.subscribe('videocallingtour/start', function() {

o = true;

p.inform('dialog/close-all');

});

j.subscribe('videocallingtour/end', function() {

o = false;

});

e.exports = p;

}, null);

\_\_d("ChatEmployeeAwayWarning", ["MercuryIDs", "MercuryParticipants", "MercuryViewer"], function(a, b, c, d, e, f, g, h, i) {

var j = {

updateEmployeeAwayWarning: function(k, l, m) {

h.get(i.getID(), function(n) {

var o = n.employee;

if (o)

h.getMulti(k.participants, function(p) {

var q = false, r = g.getUserIDFromThreadID(k.thread\_id);

if (!r)

return;

var s = 'fbid:' + r;

for (var t in p)

if (s == p[t].id && p[t].is\_employee\_away)

q = true;

if (q) {

l();

} else

m();

});

});

}

};

e.exports = j;

}, null);

\_\_d("getImageSize", ["DataViewPolyfill"], function(a, b, c, d, e, f, g) {

var h = a.DataView || g;

function i(m) {

return {

width: m.getUint16(6, true),

height: m.getUint16(8, true)

};

}

function j(m) {

return {

width: m.getUint32(16, false),

height: m.getUint32(20, false)

};

}

function k(m) {

var n = m.byteLength, o = 2;

while (o < n) {

var p = m.getUint16(o, false);

o += 2;

if (p == 65472 || p == 65474) {

return {

width: m.getUint16(o + 5, false),

height: m.getUint16(o + 3, false)

};

} else

o += m.getUint16(o, false);

}

return null;

}

function l(m) {

var n = new h(m);

if (n.getUint8(0) == 255 && n.getUint8(1) == 216)

return k(n);

if (n.getUint8(0) == 71 && n.getUint8(1) == 73 && n.getUint8(2) == 70)

return i(n);

if (n.getUint8(0) == 137 && n.getUint8(1) == 80 && n.getUint8(2) == 78 && n.getUint8(3) == 71)

return j(n);

return null;

}

e.exports = l;

l.gif = i;

l.png = j;

l.jpeg = k;

}, null);

\_\_d("ChatPhotoUploader.react", ["ChatAutoSendPhotoUploader", "Image.react", "InlineBlock.react", "Link.react", "MercuryConfig", "React", "SubscriptionsHandler", "cx", "emptyFunction", "fbt", "ix"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q) {

'use strict';

var r = l.createClass({

displayName: 'ChatPhotoUploader',

propTypes: {

actionURI: l.PropTypes.string,

onSubmit: l.PropTypes.func,

onAllUploadsComplete: l.PropTypes.func,

onLastUploadFail: l.PropTypes.func,

onLastUploadCancel: l.PropTypes.func

},

getDefaultProps: function() {

return {

actionURI: k.upload\_url,

onSubmit: o,

onAllUploadsComplete: o,

onLastUploadFail: o,

onLastUploadCancel: o

};

},

componentDidMount: function() {

this.\_uploader = new g(this.refs.form.getDOMNode(), this.refs.input.getDOMNode(), this.refs.attachID.getDOMNode());

this.\_subscriptions = new m();

this.\_subscriptions.addSubscriptions(this.\_uploader.subscribe('submit', this.\_handleSubmit), this.\_uploader.subscribe('all-uploads-completed', this.\_handleAllUploadsCompleted), this.\_uploader.subscribe('last-upload-failed', this.\_handleLastUploadFailed), this.\_uploader.subscribe('last-upload-canceled', this.\_handleLastUploadCanceled));

},

shouldComponentUpdate: function(s, t) {

return s.actionURI != this.props.actionURI;

},

componentWillUnmount: function() {

this.\_subscriptions && this.\_subscriptions.release();

this.\_uploader && this.\_uploader.destroy();

},

render: function() {

return (l.createElement("form", {

action: this.props.actionURI,

method: "post",

ref: "form"

}, l.createElement("input", {

type: "hidden",

name: "attach\_id",

ref: "attachID"

}), l.createElement("input", {

type: "hidden",

name: "images\_only",

value: true

}), l.createElement(i, {

className: "\_m \_4q60 \_3rzn"

}, l.createElement(j, {

className: "\_4q61 \_509v"

}, l.createElement(h, {

className: "\_509w",

src: q('/images/chat/chat\_camera\_icon.png'),

alt: "Kamera"

})), l.createElement("input", {

type: "file",

className: "\_n",

name: "attachment[]",

multiple: "true",

accept: "image/\*",

ref: "input"

}))));

},

\_handleSubmit: function(s, t) {

this.props.onSubmit(s, t);

},

\_handleAllUploadsCompleted: function(s, t) {

this.props.onAllUploadsComplete(s, t);

},

\_handleLastUploadFailed: function(s, t) {

this.props.onLastUploadFail(s, t);

},

\_handleLastUploadCanceled: function(s, t) {

this.props.onLastUploadCancel(s, t);

},

isUploading: function() {

return this.\_uploader && this.\_uploader.isUploading();

},

uploadFile: function(s) {

this.\_uploader.uploadFile(s);

}

});

e.exports = r;

}, null);

\_\_d("ChatPrivacyActionController", ["ChatVisibility", "JSLogger", "PresencePrivacy", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j) {

var k = function(l, m) {

this.\_userID = l;

this.\_logger = h.create('blackbird');

this.\_getState = function() {

if (g.isOnline())

return i.allows(this.\_userID) ? k.NORMAL : k.BLOCKED;

return k.OFFLINE;

};

this.\_togglePrivacy = function() {

var n = this.\_getState();

switch (this.\_getState()) {

case k.OFFLINE:

if (g.isOnline()) {

this.\_logger.error('tabs\_already\_online');

break;

}

this.\_logger.log('tabs\_go\_online', {

tab\_id: this.\_userID

});

g.goOnline(function() {

if (!i.allows(this.\_userID)) {

if (this.\_getState() !== k.BLOCKED)

this.\_logger.error('privacy\_action\_controller\_blocked\_inconsistency');

this.\_togglePrivacy();

}

}.bind(this));

break;

case k.BLOCKED:

i.allow(this.\_userID);

this.\_logger.log('tabs\_unblock', {

tab\_id: this.\_userID

});

break;

case k.NORMAL:

i.disallow(this.\_userID);

this.\_logger.log('tabs\_block', {

tab\_id: this.\_userID

});

break;

}

};

setTimeout(function() {

var n = function() {

m && m(this.\_getState());

}.bind(this);

n();

this.\_subscribeToken = i.subscribe('privacy-changed', n);

}.bind(this), 0);

};

k.OFFLINE = 'offline';

k.BLOCKED = 'blocked';

k.NORMAL = 'normal';

j(k.prototype, {

togglePrivacy: function() {

this.\_logger.log('gear\_menu\_toggle\_privacy', {

tab\_id: this.\_userID

});

this.\_togglePrivacy();

},

destroy: function() {

this.\_subscribeToken && i.unsubscribe(this.\_subscribeToken);

}

});

e.exports = k;

}, null);

\_\_d("ChatShareLinkUploader", ["AsyncRequest", "CSS", "DOM", "URLScraper", "MercuryShareLinkUploader", "MercuryAttachmentTemplates", "Event"], function(a, b, c, d, e, f, g, h, i, j, k, l, m) {

for (var n in k)

if (k.hasOwnProperty(n))

p[n] = k[n];

var o = k === null ? null: k.prototype;

p.prototype = Object.create(o);

p.prototype.constructor = p;

p.\_\_superConstructor\_\_ = k;

function p(q, r, s, t) {

"use strict";

this.$ChatShareLinkUploader0 = q;

this.$ChatShareLinkUploader1 = r;

this.$ChatShareLinkUploader2 = s;

this.$ChatShareLinkUploader3 = false;

this.$ChatShareLinkUploader4 = null;

this.$ChatShareLinkUploader5 = t;

m.listen(r, 'keyup', function() {

var u = t ? t(): r.value;

!u.length && this.$ChatShareLinkUploader6.enable();

}.bind(this));

this.$ChatShareLinkUploader6 = new j(r, t);

this.$ChatShareLinkUploader6.subscribe('match', function(u, v) {

this.loadShare(v && v.url);

}.bind(this));

this.$ChatShareLinkUploader7 = 1337;

this.$ChatShareLinkUploader8 = null;

}

p.prototype.getAttachData = function() {

"use strict";

return this.loadAttachData(this.$ChatShareLinkUploader3, this.$ChatShareLinkUploader0, this.$ChatShareLinkUploader1, this.$ChatShareLinkUploader5);

};

p.prototype.close = function() {

"use strict";

this.$ChatShareLinkUploader9();

this.$ChatShareLinkUploader6.disable();

this.inform('closed');

};

p.prototype.clear = function() {

"use strict";

this.$ChatShareLinkUploader9();

this.$ChatShareLinkUploader6.enable();

};

p.prototype.$ChatShareLinkUploader9 = function() {

"use strict";

var q = this.$ChatShareLinkUploader8;

delete this.$ChatShareLinkUploader8;

if (q) {

i.remove(q);

this.inform('dom-updated');

}

this.$ChatShareLinkUploadera();

this.$ChatShareLinkUploader3 = false;

this.$ChatShareLinkUploader4 && this.$ChatShareLinkUploader4.abort();

this.$ChatShareLinkUploader4 = null;

this.$ChatShareLinkUploader6 && this.$ChatShareLinkUploader6.reset();

};

p.prototype.loadShareFromParams = function(q, r) {

"use strict";

this.$ChatShareLinkUploaderb(new g().setMethod('POST').setURI('/ajax/share\_params.php').setData({

chat: true,

share\_type: q,

share\_params: r

}));

};

p.prototype.loadShare = function(q, r) {

"use strict";

this.$ChatShareLinkUploaderb(new g().setMethod('POST').setURI('/ajax/share\_scrape.php').setData({

chat: true,

extra\_params: JSON.stringify(r),

u: q

}));

};

p.prototype.$ChatShareLinkUploadera = function() {

"use strict";

return (++this.$ChatShareLinkUploader7);

};

p.prototype.$ChatShareLinkUploaderc = function() {

"use strict";

return this.$ChatShareLinkUploader7;

};

p.prototype.$ChatShareLinkUploaderb = function(q) {

"use strict";

this.$ChatShareLinkUploader4 && this.$ChatShareLinkUploader4.abort();

var r = l[':fb:mercury:share-link-row'].build();

if (this.$ChatShareLinkUploader8)

i.remove(this.$ChatShareLinkUploader8);

this.$ChatShareLinkUploader8 = r.getRoot();

m.listen(r.getNode('closeFileUpload'), 'click', function(event) {

this.close();

event.kill();

}.bind(this));

i.appendContent(this.$ChatShareLinkUploader0, this.$ChatShareLinkUploader8);

h.show(this.$ChatShareLinkUploader0);

this.inform('dom-updated');

this.$ChatShareLinkUploader4 = q.setHandler(this.$ChatShareLinkUploaderd.bind(this, this.$ChatShareLinkUploadera()));

this.$ChatShareLinkUploader4.send();

this.inform('link-detected');

};

p.prototype.$ChatShareLinkUploaderd = function(q, r) {

"use strict";

if (this.$ChatShareLinkUploaderc() !== q)

return;

this.$ChatShareLinkUploader4 = null;

i.appendContent(this.$ChatShareLinkUploader8, r.payload);

this.$ChatShareLinkUploader2();

this.$ChatShareLinkUploader3 = true;

if (!this.getAttachData()) {

this.close();

return;

}

h.addClass(this.$ChatShareLinkUploader8, 'done');

};

e.exports = p;

}, null);

\_\_d("StickerContextualDialog.react", ["ReactAbstractContextualDialog", "ReactLayer", "cx"], function(a, b, c, d, e, f, g, h, i) {

'use strict';

var j = 47, k = h.createClass(g.createSpec({

displayName: 'StickerXUIContextualDialog',

theme: {

wrapperClassName: "\_53ii",

arrowDimensions: {

offset: j,

length: 16

}

}

}));

k.WIDTH = {

NORMAL: 312,

WIDE: 400

};

e.exports = k;

}, null);

\_\_d("MessagesEmoticons.react", ["Grid.react", "React", "cx", "emptyFunction", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = g.GridItem, m = 8, n = {

smile: "L\u00e4cheln",

frown: "Stirnrunzeln",

tongue: "Zunge herausstrecken",

grin: "Grinsen",

gasp: "Einatmen",

wink: "Zwinkern",

pacman: "Pacman",

grumpy: "m\u00fcrrisch",

unsure: "unsicher",

cry: "Weinen",

kiki: "KiKi",

glasses: "Brille",

sunglasses: "Sonnenbrille",

heart: "Herz",

devil: "Teufel",

angel: "Engel",

squint: "Schielen",

confused: "verwirrt",

upset: "beleidigt",

colonthree: "colonthree",

like: "gef\u00e4llt mir"

}, o = h.createClass({

displayName: 'MessagesEmoticons',

propTypes: {

onEmoticonSelect: h.PropTypes.func

},

getDefaultProps: function() {

return {

onEmoticonSelect: j

};

},

getEmoticons: function() {

return Object.keys(n).map(function(p) {

return (h.createElement(l, {

key: p

}, h.createElement("div", {

className: "panelCell"

}, h.createElement("a", {

'aria-label': n[p],

className: 'emoticon emoticon\_' + p,

onClick: function() {

return this.props.onEmoticonSelect(p);

}.bind(this)

}))));

}.bind(this));

},

render: function() {

return (h.createElement("div", {

className: "emoticonsTable"

}, h.createElement(g, {

cols: m,

alignv: "middle",

alignh: "center",

spacing: "pam"

}, this.getEmoticons())));

}

});

e.exports = o;

}, null);

\_\_d("StickerActions", ["BanzaiLogger", "StickerConstants", "StickersDispatcher"], function(a, b, c, d, e, f, g, h, i) {

'use strict';

var j = {

addPack: function(k) {

i.handleUpdateFromViewActions({

actions: [{

actionType: h.ActionTypes.ADD\_PACK,

packID: k

}

]

});

},

removePack: function(k) {

i.handleUpdateFromViewActions({

actions: [{

actionType: h.ActionTypes.REMOVE\_PACK,

packID: k

}

]

});

},

selectPack: function(k, l) {

if (!l)

g.log('StickersLoggerConfig', {

event: 'select\_pack',

packid: k

});

i.handleUpdateFromViewActions({

actions: [{

actionType: h.ActionTypes.SELECT\_PACK,

packID: k

}

]

});

},

resetNumNewPacks: function() {

i.handleUpdateFromViewActions({

actions: [{

actionType: h.ActionTypes.RESET\_NUM\_NEW\_PACKS

}

]

});

}

};

e.exports = j;

}, null);

\_\_d("XStickerPackImagesControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/{pack\_id}\/images\/", {

pack\_id: {

type: "Int",

required: true

},

sticker\_size: {

type: "Int"

}

});

}, null);

\_\_d("XStickersAddPackControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/addpack\/", {

pack\_id: {

type: "Int"

},

size: {

type: "String"

},

redirect\_uri: {

type: "String"

},

is\_promoted: {

type: "Bool"

}

});

}, null);

\_\_d("XStickersRemovePackControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/removepack\/", {

pack\_id: {

type: "Int"

},

size: {

type: "String"

},

redirect\_uri: {

type: "String"

}

});

}, null);

\_\_d("XStickerStateInitialDataControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/state\/", {});

}, null);

\_\_d("XStickerStatePackDataControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/state\/pack\/", {

pack\_id: {

type: "Int",

required: true

}

});

}, null);

\_\_d("XStickerStateStoreDataControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/state\/store\/", {});

}, null);

\_\_d("XStickerSearchPromotePackControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/search\/promotePack\/", {

sticker\_id: {

type: "Int",

required: true

}

});

}, null);

\_\_d("XStickerSearchNUXSeenControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/searchNUX\/seen\/", {

tooltip\_only: {

type: "Bool"

}

});

}, null);

\_\_d("XStickerTagImagesControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/tag\/images\/", {

tag\_ids: {

type: "IntVector",

required: true

}

});

}, null);

\_\_d("XStickerTagDataControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/tag\/data\/", {});

}, null);

\_\_d("XStickerQueryImagesControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/query\/images\/", {

query: {

type: "String",

required: true

}

});

}, null);

\_\_d("StickerServerRequests", ["AsyncRequest", "Promise", "XStickerPackImagesControllerURIBuilder", "XStickersAddPackControllerURIBuilder", "XStickersRemovePackControllerURIBuilder", "XStickerStateInitialDataControllerURIBuilder", "XStickerStatePackDataControllerURIBuilder", "XStickerStateStoreDataControllerURIBuilder", "XStickerSearchPromotePackControllerURIBuilder", "XStickerSearchNUXSeenControllerURIBuilder", "XStickerTagImagesControllerURIBuilder", "XStickerTagDataControllerURIBuilder", "XStickerQueryImagesControllerURIBuilder"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s) {

'use strict';

var t = {

addPack: function(v, w, x) {

var y = new j().setInt('pack\_id', v).setBool('is\_promoted', w).getURI();

new g(y).setHandler(function(z) {

return x(z.payload);

}).send();

},

removePack: function(v, w) {

var x = new k().setInt('pack\_id', v).getURI();

new g(x).setHandler(function(y) {

return w(y.payload);

}).send();

},

getStickersForPack: function(v, w) {

var x = new i().setInt('pack\_id', v).setInt('sticker\_size', w).getURI();

return new h(function(y, z) {

return new g().setURI(x).setMethod("POST").setHandler(function(aa) {

return y(aa.payload);

}).send();

});

},

getStickersForTags: function(v, w) {

new g().setURI(new q().setIntVector('tag\_ids', v).getURI()).setMethod("POST").setHandler(function(x) {

return w(x);

}).send();

},

getStickersForQuery: function(v, w) {

new g().setURI(new s().setString('query', v).getURI()).setMethod("POST").setHandler(function(x) {

return w(x);

}).send();

},

fetchTrayData: function(v) {

return u(new l().getURI(), v, true);

},

fetchStoreData: function(v) {

return u(new n().getURI(), v);

},

fetchPackData: function(v, w) {

return u(new m().setInt('pack\_id', v).getURI(), w);

},

fetchTagData: function(v) {

return u(new r().getURI(), v, true);

},

markSeenSearchNUX: function() {

new g(new p().setBool('tooltip\_only', false).getURI().toString()).send();

},

markSeenSearchTooltipNUX: function() {

new g(new p().setBool('tooltip\_only', true).getURI().toString()).send();

},

promotePackSentFromSearch: function(v, w) {

var x = new o().setInt('sticker\_id', v).getURI();

u(x, w, true);

}

};

function u(v, w, x) {

return new h(function(y, z) {

return new g(v).setHandler(function(aa) {

return y(aa.getPayload());

}).setAllowCrossPageTransition(x).send();

}).then(w);

}

e.exports = t;

}, null);

\_\_d("StickerImages", ["StickerServerRequests"], function(a, b, c, d, e, f, g) {

'use strict';

var h = {}, i = {}, j = {}, k = {

requestStickersForTags: function(l, m) {

var n = l.filter(function(o) {

return !j[o.id];

}).map(function(o) {

return o.id;

});

if (n.length > 0) {

g.getStickersForTags(n, function(o) {

for (var p in o.payload)

j[p] = o.payload[p];

m();

});

} else

m();

},

getStickerTagsData: function() {

return j;

},

requestStickersForPack: function(l, m, n) {

if (!l)

return;

if (!i[l]) {

i[l] = g.getStickersForPack(l, m);

i[l].then(function(o) {

o.forEach(function(p) {

return this.cacheSticker(p);

}.bind(this));

}.bind(this));

}

i[l].then(function(o) {

return n(o);

});

},

cacheSticker: function(l) {

h[l.id] = l;

},

getSticker: function(l) {

return h[l];

}

};

e.exports = k;

}, null);

\_\_d("StickerState", ["ImmutableObject", "StickerConfig", "PresenceState", "StickerConstants", "StickerImages", "StickersDispatcher", "StickerServerRequests", "mixInEventEmitter"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n) {

'use strict';

var o = {}, p, q, r, s = null, t = 0, u = {}, v = [], w = [], x = [], y = null, z = i.get();

if (z)

y = z.tray\_pack\_id;

var aa = h.ShowStickerReplyNUX, ba = h.ShowStickerSearchNUXIndicator, ca = h.ShowStickerSearchNUXTooltip, da = {

onTrayDataReady: function(ga) {

if (!q)

q = m.fetchTrayData(function(ha) {

v = ha.packs.map(function(ia) {

return ea(ia);

});

r = ha.mru;

t = ha.num\_new\_packs;

x = ha.tags;

});

q.then(ga);

},

onStoreDataReady: function(ga) {

if (!p)

p = m.fetchStoreData(function(ha) {

w = ha.packs.map(function(ia) {

return ea(ia);

});

});

p.then(ga);

},

onPackDataReady: function(ga, ha) {

if (!o[ga])

o[ga] = m.fetchPackData(ga, function(ia) {

ea(ia);

});

o[ga].then(ha);

},

getPacksInTray: function() {

return v;

},

getPacksInCommentsTray: function() {

var ga = [], ha = [], ia;

for (var ja = 0; ja < v.length; ja++)

if (v[ja].isSearch) {

continue;

} else if (v[ja].id == j.EMOTICON\_PACK\_ID) {

ia = v[ja];

} else if (v[ja].isMessengerOnly) {

ha.push(v[ja]);

} else

ga.push(v[ja]);

var ka = ga.concat(ha);

if (ia)

ka.push(ia);

return ka;

},

getPackIDsInTray: function() {

return v.map(function(ga) {

return ga.id;

});

},

getPacksInStore: function() {

return w;

},

getPack: function(ga) {

return u[ga];

},

getNumNewPacks: function() {

return t;

},

getFeaturedTags: function() {

return x;

},

resetNumNewPacks: function() {

if (t) {

t = 0;

this.emit(this.NUM\_NEW\_PACKS\_CHANGED, 0);

}

},

addMRUPack: function() {

if (r) {

this.\_addPackToTray(r);

r = null;

}

},

addPack: function(ga) {

var ha = false;

if (u[ga])

ha = u[ga].isPromoted;

m.addPack(ga, ha, function(ia) {

this.\_addPackToTray(ia);

}.bind(this));

},

removePack: function(ga) {

m.removePack(ga, function(ha) {

v = v.filter(function(ia) {

return ia.id !== ha;

});

u[ha] = g.set(u[ha], {

isInTray: false

});

w = w.map(function(ia) {

return u[ia.id];

});

this.emit(this.PACKS\_CHANGED, ha);

}.bind(this));

},

setTrayPackID: function(ga) {

y = ga;

i.doSync();

this.emit(da.PACK\_SELECTED, ga);

},

getTrayPackID: function() {

return y;

},

shouldShowStickerReplyNUX: function() {

return aa;

},

clearShowStickerReplyNUX: function() {

aa = false;

},

shouldShowStickerSearchNUXIndicator: function() {

return ba;

},

shouldShowStickerSearchNUXTooltip: function() {

return ca;

},

clearShowStickerSearchNUXIndicator: function() {

m.markSeenSearchNUX();

ba = false;

},

\_addPackToTray: function(ga, ha) {

ea(ga);

u[ga.id] = g.set(u[ga.id], {

isPurchased: !ha,

isInTray: true

});

var ia = v.filter(function(ka) {

return ka.id === ga.id;

}).length > 0;

if (!ia) {

var ja;

if (!ha) {

ja = v.filter(function(ka) {

return ka.isMRU || ka.isSearch;

}).length;

} else

ja = v.filter(function(ka) {

return ka.id !== j.EMOTICON\_PACK\_ID;

}).length;

v.splice(ja, 0, u[ga.id]);

}

w = w.map(function(ka) {

return u[ka.id];

});

this.emit(this.PACKS\_CHANGED, ga.id);

},

\_addRecentlyUsedSticker: function(ga) {

for (var ha = 0; ha < s.length; ++ha)

if (s[ha].id === ga.id) {

s.splice(ha, 1);

break;

}

s.unshift(ga);

},

getMRUStickerPack: function(ga) {

if (!s)

s = ga;

return s;

},

updateRecentlyUsed: function(ga) {

if (!s) {

this.addMRUPack();

return;

}

this.\_addRecentlyUsedSticker(k.getSticker(ga));

},

promotePackSentFromSearch: function(ga) {

m.promotePackSentFromSearch(ga, function(ha) {

return ha && this.\_addPackToTray(ha, true);

}.bind(this));

}

};

l.register(function(ga) {

if (ga.actions && ga.actions.length) {

var ha = j.ActionTypes;

ga.actions.forEach(function(ia) {

switch (ia.actionType) {

case ha.ADD\_PACK:

da.addPack(ia.packID);

break;

case ha.REMOVE\_PACK:

da.removePack(ia.packID);

break;

case ha.SELECT\_PACK:

da.setTrayPackID(ia.packID);

break;

case ha.RESET\_NUM\_NEW\_PACKS:

da.resetNumNewPacks();

break;

}

});

}

});

da.PACKS\_CHANGED = 'StickerState/packsChanged';

da.PACK\_SELECTED = 'StickerState/packSelected';

da.NUM\_NEW\_PACKS\_CHANGED = 'StickerState/numNewPacksChanged';

i.registerStateStorer(function(ga) {

ga.tray\_pack\_id = y;

});

function ea(ga) {

var ha = ga.id;

if (!u[ha]) {

return (u[ha] = new g(ga));

} else

return (u[ha] = g.set(u[ha], ga));

}

var fa = {};

fa[da.PACK\_SELECTED] = true;

fa[da.PACKS\_CHANGED] = true;

fa[da.NUM\_NEW\_PACKS\_CHANGED] = true;

n(da, fa);

e.exports = da;

}, null);

\_\_d("DialogFitHeight", ["CSS", "DOM", "Event", "Style", "SubscriptionsHandler", "Vector", "csx", "cx", "throttle"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o) {

var p = 450, q = 100, r = 67, s = 67;

function t(u) {

"use strict";

this.$DialogFitHeight0 = u;

}

t.prototype.enable = function() {

"use strict";

this.$DialogFitHeight1 = new k();

this.$DialogFitHeight1.addSubscriptions(this.$DialogFitHeight0.subscribe('beforeshow', this.$DialogFitHeight2.bind(this)), i.listen(window, 'resize', o(this.$DialogFitHeight2.bind(this))));

this.$DialogFitHeight3 = h.find(this.$DialogFitHeight0.getRoot(), ".\_4-i2");

g.addClass(this.$DialogFitHeight3, "\_5pfh");

this.$DialogFitHeight4 = q;

if (h.scry(this.$DialogFitHeight0.getRoot(), ".\_4-i0").length)

this.$DialogFitHeight4 += r;

if (h.scry(this.$DialogFitHeight0.getRoot(), ".\_5a8u").length)

this.$DialogFitHeight4 += s;

};

t.prototype.disable = function() {

"use strict";

this.$DialogFitHeight1.release();

this.$DialogFitHeight1 = null;

g.removeClass(this.$DialogFitHeight3, "\_5pfh");

};

t.prototype.$DialogFitHeight2 = function() {

"use strict";

var u = l.getViewportDimensions().y, v = u - this.$DialogFitHeight4;

j.set(this.$DialogFitHeight3, 'height', Math.max(p, v) + 'px');

this.$DialogFitHeight0.updatePosition();

};

e.exports = t;

}, null);

\_\_d("StickerStoreController", ["Bootloader", "DialogFitHeight", "DOM", "LayerAutoFocus", "LayerFadeOnHide", "LayerHideOnEscape", "React", "XUIDialog.react", "XUIDialogBody.react", "XUISpinner.react", "cx", "requestAnimationFrame"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r) {

'use strict';

var s = 688, t = null, u = m.createClass({

displayName: 'StoreLayer',

getInitialState: function() {

return {

renderStore: function() {

return (m.createElement("div", {

className: "\_5r5e"

}, m.createElement(p, {

background: "light",

size: "large"

})));

}

};

},

componentDidMount: function() {

g.loadModules(["StickerStore.react"], function(x) {

this.setState({

renderStore: function() {

return (m.createElement(x, {

currentPackID: this.props.packID,

onPackSelect: this.\_packSelected,

shown: this.props.shown

}));

}.bind(this)

});

}.bind(this));

},

\_packSelected: function(x) {

this.setProps({

packID: x

});

},

\_onToggle: function(x) {

r(this.setProps.bind(this, {

shown: x

}, null));

},

render: function() {

return (m.createElement(n, {

behaviors: {

DialogFitHeight: h,

LayerAutoFocus: j,

LayerFadeOnHide: k,

LayerHideOnEscape: l

},

onToggle: this.\_onToggle,

shown: this.props.shown,

width: s

}, m.createElement(o, {

className: "\_5rq- autofocus"

}, this.state.renderStore())));

}

});

function v(x) {

var y = i.create('div');

i.appendContent(document.body, y);

t = m.render(m.createElement(u, {

packID: x,

shown: true

}), y);

}

var w = {

showStore: function(x) {

if (!t) {

v(x);

} else

t.setProps({

shown: true,

packID: x

});

}

};

e.exports = w;

}, null);

\_\_d("StickersFlyoutPackSelector.react", ["Animation", "ImmutableObject", "Locale", "React", "Image.react", "StickerActions", "StickerConfig", "StickerConstants", "StickerState", "StickerStoreController", "XUIBadge.react", "cx", "emptyFunction", "fbt", "ix"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u) {

'use strict';

var v = 4, w = 200, x = "Sticker Store", y = j.createClass({

displayName: 'StickersFlyoutPackSelector',

propTypes: {

isComments: j.PropTypes.bool,

numNewPacks: j.PropTypes.number,

packs: j.PropTypes.arrayOf(j.PropTypes.instanceOf(h)).isRequired,

onPackClick: j.PropTypes.func,

selectedPackID: j.PropTypes.string,

resetTagSelectorFunc: j.PropTypes.func

},

getInitialState: function() {

return {

animating: false,

page: 0

};

},

getDefaultProps: function() {

return {

isComments: false,

numNewPacks: 0

};

},

shouldComponentUpdate: function(aa, ba) {

return !ba.animating;

},

onFlyoutShown: function() {

if (this.props.packs.length > 0) {

var aa = this.\_calculatePageForPack(this.props.selectedPackID);

if (this.state.page !== aa)

this.\_setPage(aa, 0);

}

},

\_calculatePageForPack: function(aa) {

for (var ba = 0; ba < this.props.packs.length; ba++)

if (this.props.packs[ba].id == aa)

return ba <= v ? 0 : Math.floor((ba - 1) / v);

return 0;

},

\_setPage: function(aa, ba) {

if (this.state.animating)

return;

this.setState({

animating: true,

page: aa

}, function() {

var ca = this.refs.positioner.getDOMNode(), da = this.\_calculatePosition(aa);

new g(ca).to(da.reference, da.offset + 'px').ondone(function() {

return this.setState({

animating: false

});

}.bind(this)).duration(ba).go();

});

},

\_calculatePosition: function(aa) {

var ba = this.refs.positioner.getDOMNode(), ca = ba.childNodes[aa].offsetLeft;

if (i.isRTL()) {

var da = ba.offsetWidth, ea = ba.childNodes[aa].offsetWidth;

return {

reference: 'right',

offset: ca + ea - da

};

}

return {

reference: 'left',

offset: - ca

};

},

\_numPages: function() {

return Math.max(1, Math.ceil((this.props.packs.length - 1) / v));

},

\_canGoPrev: function() {

return this.state.page > 0;

},

\_canGoNext: function() {

return this.state.page + 1 < this.\_numPages();

},

\_goPrev: function() {

this.\_canGoPrev() && this.\_setPage(this.state.page - 1, w);

},

\_goNext: function() {

this.\_canGoNext() && this.\_setPage(this.state.page + 1, w);

},

\_openStore: function() {

l.resetNumNewPacks();

p.showStore();

},

render: function() {

return (j.createElement("div", {

className: "\_5r85"

}, this.\_renderStoreButton(), this.\_renderPrevArrow(), this.\_renderNextArrow(), j.createElement("div", {

className: "\_5r88"

}, j.createElement("div", {

className: "\_5r89",

ref: "positioner"

}, this.\_renderPages()))));

},

\_selectPack: function(aa) {

var ba = o.getPack(aa);

if (ba && ba.isPromoted)

l.addPack(aa);

if (aa === n.SEARCH\_PACK\_ID)

this.props.resetTagSelectorFunc();

if (this.props.onPackClick) {

this.props.onPackClick(aa);

} else

l.selectPack(aa);

},

\_renderPages: function() {

var aa = this.props.packs.map(function(ea, fa) {

return j.createElement(z, {

key: ea.id,

onClick: function() {

return this.\_selectPack(ea.id);

}.bind(this),

pack: ea,

selected: this.props.selectedPackID === ea.id,

index: fa,

isComments: this.props.isComments

});

}.bind(this)), ba = [];

for (var ca = 0; ca < aa.length; ca += v) {

var da = ca;

ca === 0 && ca++;

ba.push(j.createElement("div", {

className: "\_5r81",

key: ca

}, aa.slice(da, ca + v)));

}

return ba;

},

\_renderStoreButton: function() {

return (j.createElement("a", {

'aria-label': x,

className: "\_5r86 rfloat",

'data-hover': "tooltip",

onClick: this.\_openStore

}, j.createElement(k, {

className: "\_5r87",

src: u('/images/messaging/stickers/selector/sticker\_store.png')

}), this.\_renderJewel()));

},

\_renderJewel: function() {

var aa = this.props.numNewPacks;

if (!aa)

return null;

return (j.createElement(q, {

className: "rfloat \_3fhs",

count: aa,

maxcount: 9,

type: "special"

}));

},

\_renderPrevArrow: function() {

if (!this.\_canGoPrev())

return null;

var aa = i.isRTL() ? u('/images/messaging/stickers/selector/right.png'): u('/images/messaging/stickers/selector/left.png');

return (j.createElement("a", {

className: (("\_37wu") + (' ' + "lfloat")),

onClick: this.\_goPrev

}, j.createElement(k, {

className: "\_5r84",

src: aa

})));

},

\_renderNextArrow: function() {

if (!this.\_canGoNext())

return null;

var aa = i.isRTL() ? u('/images/messaging/stickers/selector/left.png'): u('/images/messaging/stickers/selector/right.png');

return (j.createElement("a", {

className: (("\_37wv") + (' ' + "rfloat")),

onClick: this.\_goNext

}, j.createElement(k, {

className: "\_5r84",

src: aa

})));

}

}), z = j.createClass({

displayName: 'PackIcon',

propTypes: {

index: j.PropTypes.number,

isComments: j.PropTypes.bool,

onClick: j.PropTypes.func,

pack: j.PropTypes.instanceOf(h).isRequired,

selected: j.PropTypes.bool

},

getDefaultProps: function() {

return {

isComments: false,

onClick: s

};

},

\_getPackIcon: function(aa) {

if (aa.id == n.SEARCH\_PACK\_ID)

return u('/images/messaging/stickers/icons/search.png');

if (aa.id == n.MRU\_STICKER\_PACK)

return u('/images/messaging/stickers/icons/recent.png');

if (aa.id == n.EMOTICON\_PACK\_ID)

return u('/images/messaging/stickers/icons/emoji.png');

return aa.icon;

},

render: function() {

var aa = this.props.pack, ba = m.WebStickerSearch&&!m.StickerSearchInRecent ? this.props.index === 1 || this.props.index === 2: this.props.index === 1, ca = (("\_5r8a") + (this.props.selected ? ' ' + "\_5r8b" : '') + (aa.id == n.MRU\_STICKER\_PACK ? ' ' + "\_5qcj" : '') + (aa.id == n.SEARCH\_PACK\_ID ? ' ' + "\_5qck" : '') + (ba ? ' ' + "\_eb3" : '')), da = this.props.isComments && aa.isMessengerOnly, ea = "Dieses Stickerpack ist nur in den Nachrichten verf\u00fcgbar", fa = da ? s: function() {

return this.props.onClick(aa.id);

}.bind(this);

return (j.createElement("a", {

'aria-label': da ? ea: aa.name,

className: ca,

'data-id': aa.id,

'data-hover': "tooltip",

ref: "search\_icon",

onClick: fa,

tabIndex: "0"

}, j.createElement(k, {

className: ((da ? "\_2ji6" : '') + (' ' + "\_5r8c") + (m.WebStickerSearch ? ' ' + "\_1viy" : '')),

src: this.\_getPackIcon(aa)

})));

}

});

e.exports = y;

}, null);

\_\_d("StickerSearch", ["StickerImages", "StickerServerRequests", "StickerState", "createObjectFrom", "getObjectValues"], function(a, b, c, d, e, f, g, h, i, j, k) {

'use strict';

var l = 100, m = 1000, n = 10000, o, p, q, r = null, s = {}, t = {

requestStickersForTags: function(u, v, w, x) {

p = w;

var y = u, z = v.length > 1;

if (z)

y = y.concat(v);

g.requestStickersForTags(y, function() {

if (p === w)

x(this.\_getSortedStickersForTags(u, v, w, z));

}.bind(this));

},

requestStickersForQuery: function(u, v) {

p = u;

h.getStickersForQuery(u, function(w) {

return p === u && v(w.payload);

});

},

\_getSortedStickersForTags: function(u, v, w, x) {

var y = {}, z = this.\_getTypeaheadStickers(u, w, y), aa = k(z);

if (x) {

var ba = this.\_getIntersectionStickers(v, y);

aa = aa.concat(ba.filter(function(ca) {

return !z.hasOwnProperty(ca.id);

}));

}

aa.map(function(ca) {

return g.cacheSticker(ca);

});

aa.sort(function(ca, da) {

return y[da.id] - y[ca.id];

});

return aa;

},

\_getTypeaheadStickers: function(u, v, w) {

var x = {}, y = g.getStickerTagsData();

o = j(i.getPackIDsInTray());

u.forEach(function(z) {

y[z.id].forEach(function(aa) {

var ba = aa.score + l;

if (o[aa.packID])

ba += m;

if (z.name === v)

ba += n;

if (!x[aa.id] || ba > w[aa.id]) {

x[aa.id] = aa;

w[aa.id] = ba;

}

});

});

return x;

},

\_getIntersectionStickers: function(u, v) {

var w = g.getStickerTagsData(), x = u.map(function(z) {

return w[z.id].map(function(aa) {

return aa.id;

});

}), y = w[u[0].id].filter(function(z) {

v[z.id] = z.score;

for (var aa = 1; aa < x.length; aa++)

if (x[aa].indexOf(z.id)===-1) {

return false;

} else

v[z.id] += w[u[aa].id].score;

return true;

});

return y;

},

prepareTagsData: function() {

if (!q)

q = h.fetchTagData(function(u) {

r = u;

r.map(function(v) {

return s[v.name] = v;

});

});

return q;

},

getTagByName: function(u) {

return s[u];

},

getTags: function() {

return r;

},

getTagsIndex: function() {

return s;

}

};

e.exports = t;

}, null);

\_\_d("StickerUtils", [], function(a, b, c, d, e, f) {

var g = 64, h = {

getScaledDimensions: function(i, j) {

var k, l, m;

if (j > i) {

m = g / j;

k = i \* m;

l = j \* m;

} else {

m = g / i;

k = i \* m;

l = j \* m;

}

return {

height: Math.round(k),

width: Math.round(l)

};

},

capitalizeWords: function(i) {

var j = i.split(" ");

for (var k = 0; k < j.length; k++) {

var l = j[k].charAt(0).toUpperCase();

j[k] = l + j[k].substr(1);

}

return j.join(" ");

}

};

e.exports = h;

}, null);

\_\_d("StickersFlyoutStickerSelector.react", ["BanzaiLogger", "Grid.react", "Image.react", "LegacyScrollableArea.react", "React", "Sticker.react", "StickerConstants", "StickerConfig", "StickerImages", "StickerSearch", "StickerState", "StickerUtils", "XUISpinner.react", "areEqual", "cx", "debounce", "emptyFunction", "fbt", "ix", "throttle"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z) {

'use strict';

var aa = h.GridItem, ba = 320, ca = 278, da = m.MRU\_STICKER\_PACK, ea = 64, fa = 44, ga = 112, ha = n.AutoAnimateStickerTray ? 'load\_and\_hover': 'hover', ia = v(function(ka, la) {

if (!ka)

return;

g.log('StickersLoggerConfig', {

event: 'search\_sticker',

searchtoken: ka,

numsearchresults: la.length

});

}, 1000), ja = k.createClass({

displayName: 'StickersFlyoutStickerSelector',

propTypes: {

height: k.PropTypes.number,

onScroll: k.PropTypes.func,

packID: k.PropTypes.string,

userInput: k.PropTypes.string,

typeaheadTags: k.PropTypes.array,

singleWordTags: k.PropTypes.array

},

getDefaultProps: function() {

return {

isComments: false,

onScroll: w,

shown: false

};

},

getInitialState: function() {

return {

loading: false,

stickers: []

};

},

componentDidMount: function() {

if (this.props.packID) {

this.requestStickersForPack(this.props.packID);

return;

}

if ((this.props.typeaheadTags && this.props.typeaheadTags.length > 0) || (this.props.singleWordTags && this.props.singleWordTags.length > 0)) {

this.requestStickersForTags(this.props.typeaheadTags, this.props.singleWordTags, this.props.userInput);

return;

}

if (this.props.userInput)

this.requestStickersForQuery(this.props.userInput);

},

componentWillReceiveProps: function(ka) {

if (ka.packID && ka.packID !== this.props.packID) {

this.requestStickersForPack(ka.packID);

} else if (!t(ka.typeaheadTags, this.props.typeaheadTags) ||!t(ka.singleWordTags, this.props.singleWordTags)) {

this.requestStickersForTags(ka.typeaheadTags, ka.singleWordTags, ka.userInput);

} else if (!p.getTags() && ka.userInput !== this.props.userInput)

this.requestStickersForQuery(ka.userInput);

},

requestStickersForTags: function(ka, la, ma) {

this.setState({

loading: true

});

p.requestStickersForTags(ka, la, ma, function(na) {

if (!this.props.tagChosen)

ia(ma, na);

if (this.props.isComments)

na = na.filter(function(oa) {

return !oa.isMessengerOnly;

});

this.isMounted() && this.setState({

loading: false,

stickers: na.slice(0, 40)

});

}.bind(this));

},

requestStickersForQuery: function(ka) {

this.setState({

loading: true

});

p.requestStickersForQuery(ka, function(la) {

la.map(function(ma) {

return o.cacheSticker(ma);

});

if (!this.props.tagChosen)

ia(ka, la);

if (this.props.isComments)

la = la.filter(function(ma) {

return !ma.isMessengerOnly;

});

this.isMounted() && this.setState({

loading: false,

stickers: la

});

}.bind(this));

},

requestStickersForPack: function(ka) {

this.setState({

loading: true

});

o.requestStickersForPack(ka, ea, function(la) {

if (ka == da) {

la = q.getMRUStickerPack(la);

if (this.props.isComments)

la = la.filter(function(ma) {

return !ma.isMessengerOnly;

});

}

this.setState({

loading: false,

stickers: la

});

}.bind(this));

},

renderStickers: function() {

return this.state.stickers.map(function(ka) {

var la = r.getScaledDimensions(ka.height, ka.width);

return (k.createElement(aa, {

key: ka.id

}, k.createElement("div", {

className: "\_5r8h",

'data-id': ka.id

}, k.createElement(l, {

animationTrigger: ha,

className: "\_5r8i",

frameCount: ka.frameCount,

frameRate: ka.frameRate || 83,

framesPerCol: ka.framesPerCol,

framesPerRow: ka.framesPerRow,

shown: this.props.shown,

sourceHeight: la.height,

sourceURI: ka.sourceURI,

sourceWidth: la.width,

spriteURI: ka.spriteURI,

paddedSpriteURI: ka.paddedSpriteURI,

stickerID: ka.id,

style: {

cursor: 'pointer'

}

}))));

}.bind(this));

},

\_onScroll: function() {

var ka = this.refs.stickerScrollable;

if (ka) {

var la = ka.getArea().getScrollTop();

this.props.onScroll(la);

}

},

render: function() {

if (this.state.loading) {

return (k.createElement("div", {

className: "\_e0r"

}, k.createElement(s, {

size: "large"

})));

} else if (this.state.stickers.length === 0) {

var ka = (ba - fa \* 2 - ga) / 2 + "px";

return (k.createElement("div", {

className: "\_5jdt",

style: {

marginTop: ka

}

}, k.createElement(i, {

src: y("/images/messaging/stickers/icons/sad\_face.png")

}), k.createElement("p", null, "Keine Sticker vorhanden")));

}

return (k.createElement(j, {

ref: "stickerScrollable",

height: this.props.height || ba,

onScroll: z(this.\_onScroll, 200),

width: ca,

fade: true

}, k.createElement("div", {

className: "\_5r8k"

}, k.createElement(h, {

cols: 4,

fixed: true

}, this.renderStickers()))));

}

});

e.exports = ja;

}, null);

\_\_d("StickersFlyoutTagSelector.react", ["BanzaiLogger", "Grid.react", "LegacyScrollableArea.react", "Parent", "React", "Image.react", "StickerConfig", "StickerConstants", "StickersFlyoutStickerSelector.react", "StickerSearch", "StickerState", "StickerUtils", "Toggler", "XUIButton.react", "XUICloseButton.react", "XUITextInput.react", "cx", "emptyFunction", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y) {

'use strict';

var z = h.GridItem, aa = 320, ba = 278, ca = 44, da = k.createClass({

displayName: 'StickersFlyoutTagSelector',

propTypes: {

trigger: k.PropTypes.string,

isComments: k.PropTypes.bool,

resetTrigger: k.PropTypes.func,

shown: k.PropTypes.bool

},

getInitialState: function() {

return {

input: '',

isScrolling: false,

tagChosen: false,

typeaheadTags: [],

singleWordTags: []

};

},

getDefaultProps: function() {

return {

trigger: null,

isComments: false,

resetTrigger: x,

shown: false

};

},

componentDidMount: function() {

var ea = j.byClass(this.getDOMNode(), 'uiToggle');

if (ea)

this.\_togglerSub = s.listen('show', ea, function() {

this.isMounted() && this.focusInput();

}.bind(this));

this.focusInput();

this.selectTrigger(this.props.trigger);

},

componentWillUnmount: function() {

this.\_togglerSub && this.\_togglerSub.unsubscribe();

},

componentWillReceiveProps: function(ea) {

this.selectTrigger(ea.trigger);

},

focusInput: function() {

this.refs.inputField.focusInput && this.refs.inputField.focusInput();

},

\_inputChanged: function(ea) {

var fa = this.\_normalizeInput(ea.target.value), ga = p.getTags();

if (ga) {

var ha = ga.filter(function(ka) {

return ka.name.indexOf(fa) === 0;

}), ia = fa.split(/\s+/), ja = ia.map(function(ka) {

return p.getTagByName(ka);

}).filter(function(ka) {

return ka !== undefined;

});

if (ja.length !== ia.length)

ja = [];

this.setState({

input: ea.target.value,

tagChosen: false,

typeaheadTags: ha,

singleWordTags: ja,

isScrolling: false

});

} else

this.setState({

input: ea.target.value,

tagChosen: false,

isScrolling: false

});

},

\_normalizeInput: function(ea) {

return ea.trim().replace(/\s+/, ' ').toLowerCase();

},

\_handleResetButtonClick: function(ea) {

ea.preventDefault();

this.reset();

},

reset: function() {

this.setState(this.getInitialState());

},

renderContentArea: function() {

if (this.state.tagChosen || this.\_normalizeInput(this.state.input).length > 1) {

return this.renderStickers();

} else

return m.StickerSearchInRecent ? this.renderRecentStickers() : this.renderTags();

},

render: function() {

return (k.createElement("div", {

className: (("\_217a") + (this.state.isScrolling ? ' ' + "\_1hg1" : ''))

}, k.createElement("div", {

className: "\_5jdr"

}, k.createElement("span", {

className: "\_5jds"

}), k.createElement(v, {

onChange: this.\_inputChanged,

ref: "inputField",

placeholder: "Suche nach Stickern",

value: this.state.input

}), k.createElement(u, {

size: "small",

onClick: this.\_handleResetButtonClick,

className: ((this.state.input.length === 0 ? "hidden\_elem" : ''))

})), this.renderContentArea()));

},

selectTag: function(ea) {

g.log('StickersLoggerConfig', {

event: 'select\_tag',

tagid: ea.id

});

this.setState({

tagChosen: true,

typeaheadTags: [ea],

singleWordTags: [],

input: r.capitalizeWords(ea.name)

});

this.focusInput();

},

selectTrigger: function(ea) {

if (ea !== null) {

var fa = p.getTagByName(ea);

this.setState({

tagChosen: true,

typeaheadTags: [fa],

singleWordTags: [],

input: r.capitalizeWords(fa.name)

});

this.props.resetTrigger();

}

},

\_onScroll: function(ea) {

this.setState({

isScrolling: !!ea

});

},

renderStickers: function() {

var ea = this.\_normalizeInput(this.state.input);

return (k.createElement(o, {

ref: "selector",

height: aa - ca,

userInput: ea,

typeaheadTags: this.state.typeaheadTags,

singleWordTags: this.state.singleWordTags,

onScroll: this.\_onScroll,

isComments: this.props.isComments,

tagChosen: this.state.tagChosen,

shown: this.props.shown

}));

},

renderRecentStickers: function() {

if (this.\_normalizeInput(this.state.input).length === 1)

return k.createElement("div", null);

return (k.createElement(o, {

ref: "selector",

height: aa - ca,

packID: n.MRU\_STICKER\_PACK,

onScroll: this.\_onScroll,

isComments: this.props.isComments,

shown: this.props.shown

}));

},

renderTags: function() {

var ea = q.getFeaturedTags().filter(function(fa) {

return fa.sourceURI !== null;

}).sort(function(fa, ga) {

return fa.order - ga.order;

}).map(function(fa, ga) {

return k.createElement(z, {

key: ga

}, k.createElement("div", {

className: (("\_t5c") + (ga < 2 ? ' ' + "\_1b27" : '') + (ga%2 === 0 ? ' ' + "\_t5d" : '') + (ga%2 !== 0 ? ' ' + "\_t5e" : ''))

}, k.createElement(t, {

image: k.createElement(l, {

src: fa.sourceURI

}),

label: fa.name,

onClick: this.selectTag.bind(this, fa),

className: "\_5jdu",

style: {

background: '#' + fa.color\_code

},

disabled: this.\_normalizeInput(this.state.input).length === 1

})));

}.bind(this));

return (k.createElement(i, {

height: aa - ca,

width: ba - 16,

shadow: true,

fade: true,

className: "\_5jei"

}, k.createElement(h, {

spacing: "pas",

cols: 2,

fixed: true,

ref: "grid"

}, ea)));

}

});

e.exports = da;

}, null);

\_\_d("ChatStickerButton.react", ["BanzaiODS", "React", "ReactLayeredComponentMixin", "StickerContextualDialog.react", "StickersFlyout.react", "StickerState", "cx", "emptyFunction", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o) {

'use strict';

var p = 278, q = h.createClass({

displayName: 'ChatStickerButton',

propTypes: {

onStickerSelect: h.PropTypes.func,

onEmoticonSelect: h.PropTypes.func,

onFlyoutShown: h.PropTypes.func,

onFlyoutHidden: h.PropTypes.func,

packID: h.PropTypes.string

},

getInitialState: function() {

return {

shown: false

};

},

mixins: [i],

getDefaultProps: function() {

return {

onStickerSelect: n,

onEmoticonSelect: n

};

},

componentDidMount: function() {

g.bumpEntityKey('chat.web', 'sticker\_button.mounted');

this.\_subscription = l.addListener(l.PACK\_SELECTED, function(r) {

return this.setProps({

packID: r

});

}.bind(this));

},

shouldComponentUpdate: function(r, s) {

return (r.packID != this.props.packID || s.shown != this.state.shown);

},

componentWillUnmount: function() {

this.\_subscription && this.\_subscription.remove();

},

render: function() {

return (h.createElement("a", {

className: "\_4vuh",

onClick: this.showFlyout,

onMouseDown: this.\_prepareForClick,

title: "W\u00e4hle einen Sticker oder ein Emoticon aus",

ref: "link"

}, h.createElement("div", {

className: (("\_4vui") + (this.state.shown ? ' ' + "open" : ''))

})));

},

renderLayers: function() {

return {

contextualDialog: h.createElement(j, {

alignment: "left",

className: "\_5e-r",

contextRef: "link",

onBlur: this.\_hideFlyout,

position: "above",

shown: this.state.shown,

width: p

}, h.createElement("div", null, h.createElement(k, {

onShown: function(r) {

return this.props.onFlyoutShown && this.props.onFlyoutShown(r);

}.bind(this),

onHidden: function() {

return this.props.onFlyoutHidden && this.props.onFlyoutHidden();

}.bind(this),

onEscKeyDown: this.\_hideFlyout,

onStickerSelect: this.\_handleStickerSelected,

onEmoticonSelect: this.\_handleEmoticonSelected,

packID: this.props.packID,

shown: this.state.shown

})))

};

},

\_prepareForClick: function() {

this.\_clickGuard = this.state.shown;

},

showFlyout: function() {

!this.\_clickGuard && this.setState({

shown: true

});

},

\_hideFlyout: function() {

this.setState({

shown: false

});

},

\_handleStickerSelected: function(r) {

this.props.onStickerSelect(r);

},

\_handleEmoticonSelected: function(r) {

this.\_hideFlyout();

this.props.onEmoticonSelect(r);

}

});

e.exports = q;

}, null);

\_\_d("StickersFlyoutTriggerSelector.react", ["React", "Sticker.react", "StickerUtils", "XUIContextualDialog.react", "XUICloseButton.react", "cx"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

'use strict';

var m = 84, n = g.createClass({

displayName: 'StickersFlyoutTriggerSelector',

getDefaultProps: function() {

return {

stickers: []

};

},

propTypes: {

context: g.PropTypes.object.isRequired,

stickers: g.PropTypes.array,

onStickerSelect: g.PropTypes.func.isRequired

},

shouldComponentUpdate: function(o) {

if (this.props.stickers.length !== o.stickers.length)

return true;

for (var p = 0; p < Math.min(this.props.stickers.length, 5); p++)

if (this.props.stickers[p].id !== o.stickers[p].id)

return true;

return false;

},

\_renderStickers: function() {

if (!this.props.stickers || this.props.stickers.length === 0)

return null;

var o = Math.floor(Math.random() \* Math.min(this.props.stickers.length, 5)), p = this.props.stickers[o], q = i.getScaledDimensions(p.height, p.width);

return (g.createElement("div", null, g.createElement(k, {

size: "small",

className: "\_2ohj",

onClick: this.props.onDismissTrigger

}), g.createElement("div", {

className: "\_5r8h \_onc",

'data-id': p.id,

onClick: function() {

return this.props.onStickerSelect(p.id);

}.bind(this)

}, g.createElement(h, {

animationTrigger: "hover",

className: "\_5r8i",

frameCount: p.frameCount,

frameRate: p.frameRate || 83,

framesPerCol: p.framesPerCol,

framesPerRow: p.framesPerRow,

sourceHeight: q.height,

sourceURI: p.sourceURI,

sourceWidth: q.width,

spriteURI: p.spriteURI,

paddedSpriteURI: p.paddedSpriteURI,

stickerID: p.id,

style: {

cursor: 'pointer',

height: '64px'

}

}))));

},

render: function() {

return (g.createElement(j, {

alignment: "left",

autoFocus: false,

context: this.props.context,

onBlur: this.\_resetFlyout,

offsetY: 11,

position: "above",

shown: this.props.stickers && this.props.stickers.length > 0,

width: m

}, this.\_renderStickers()));

}

});

e.exports = n;

}, null);

\_\_d("StickerTrigger", ["EmoticonsList", "Map", "StickerConfig", "StickerSearch", "StickerServerRequests"], function(a, b, c, d, e, f, g, h, i, j, k) {

'use strict';

var l = i.StickerTriggerThreshold, m = new h(), n = {

findTriggerForInput: function(o, p) {

if (o === null) {

p([], null);

return;

}

j.prepareTagsData().then(function() {

return this.findTagForInput(o, function(q, r) {

return p(q, r);

});

}.bind(this));

},

findTagForInput: function(o, p) {

var q = this.findTriggeredWord(o);

if (q) {

if (m.has(q)) {

p(m.get(q), q);

} else

k.getStickersForQuery(q, function(r) {

var s = r.getPayload();

m.set(q, s);

p(s, q);

});

} else

p([], null);

},

\_getRegex: function() {

if (i.TriggerConfidence === 'emoji\_only') {

return g.noncapturingRegexp;

} else if (i.TriggerConfidence === 'high')

return (/^(?:\w+\s?){1,4}$/);

return null;

},

findTriggeredWord: function(o) {

var p = j.getTagsIndex();

if (!p)

return null;

o = o.toLowerCase();

var q = o.match(this.\_getRegex()), r = l, s = null;

for (var t = 0; q !== null && t < q.length; t++)

if (p[q[t]] && p[q[t]].sticker\_count > r) {

r = p[q[t]].sticker\_count;

s = q[t];

}

return s;

}

};

e.exports = n;

}, null);

\_\_d("StickerTriggerIndicator.react", ["React", "cx"], function(a, b, c, d, e, f, g, h) {

'use strict';

var i = g.createClass({

displayName: 'StickerTriggerIndicator',

getDefaultProps: function() {

return {

stickers: []

};

},

propTypes: {

stickers: g.PropTypes.array

},

shouldComponentUpdate: function(j) {

var k = this.props.stickers.length > 0, l = j.stickers.length > 0;

return k !== l;

},

render: function() {

var j = this.props.stickers && this.props.stickers.length > 0;

return g.createElement("span", {

className: ((j ? "emoticonIndicator" : ''))

});

}

});

e.exports = i;

}, null);

\_\_d("ChatStickerTriggerController", ["BanzaiLogger", "React", "StickerConfig", "StickersFlyoutTriggerSelector.react", "StickerTrigger", "StickerTriggerIndicator.react"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

'use strict';

function m(n, o, p, q, r) {

this.$ChatStickerTriggerController0 = null;

this.$ChatStickerTriggerController1 = null;

this.$ChatStickerTriggerController2 = null;

this.$ChatStickerTriggerController3 = r;

this.$ChatStickerTriggerController4 = q;

this.$ChatStickerTriggerController5 = null;

if (i.TriggerInterface === 'one\_tap') {

this.$ChatStickerTriggerController1 = h.render(h.createElement(j, {

context: p,

onStickerSelect: this.$ChatStickerTriggerController6.bind(this),

onDismissTrigger: this.$ChatStickerTriggerController7.bind(this)

}), n);

this.$ChatStickerTriggerController2 = n;

} else if (i.TriggerInterface === 'hyperlink') {

this.$ChatStickerTriggerController1 = h.render(h.createElement(l, null), o);

this.$ChatStickerTriggerController2 = o;

}

}

m.prototype.setInput = function(n) {

this.$ChatStickerTriggerController8 = n;

};

m.prototype.activateAfterparty = function() {

this.$ChatStickerTriggerController9 = true;

clearTimeout(this.$ChatStickerTriggerControllera);

this.$ChatStickerTriggerControllera = setTimeout(this.$ChatStickerTriggerControllerb.bind(this), 5000);

};

m.prototype.updateInput = function(n) {

if (this.$ChatStickerTriggerController9)

this.$ChatStickerTriggerControllerb();

this.$ChatStickerTriggerControllerc(n);

};

m.prototype.onFlyoutShown = function(n) {

if (this.$ChatStickerTriggerController0&&!this.$ChatStickerTriggerControllerd) {

this.$ChatStickerTriggerControllerd = true;

this.$ChatStickerTriggerController3(this.$ChatStickerTriggerController0, n);

this.$ChatStickerTriggerController1.isMounted() && this.$ChatStickerTriggerController1.setProps({

stickers: []

});

} else

n();

};

m.prototype.getTriggeredWord = function() {

return this.$ChatStickerTriggerController0;

};

m.prototype.$ChatStickerTriggerController6 = function(n) {

this.$ChatStickerTriggerController4(n);

this.$ChatStickerTriggerController7();

};

m.prototype.$ChatStickerTriggerController7 = function() {

this.$ChatStickerTriggerControllerd = true;

this.$ChatStickerTriggerController1.isMounted() && this.$ChatStickerTriggerController1.setProps({

stickers: []

});

};

m.prototype.$ChatStickerTriggerControllere = function(n, o) {

if (this.$ChatStickerTriggerController5 === null || this.$ChatStickerTriggerController5 && this.$ChatStickerTriggerController5.toLowerCase().indexOf(o)===-1) {

this.$ChatStickerTriggerController8.isMounted() && this.$ChatStickerTriggerController8.setProps({

triggeredWord: null

});

this.$ChatStickerTriggerController1.isMounted() && this.$ChatStickerTriggerController1.setProps({

stickers: []

});

return;

}

if (o)

g.log('StickersLoggerConfig', {

event: 'sticker\_trigger\_activated',

numsearchresults: n.length,

triggeredword: o,

triggerused: false

});

this.$ChatStickerTriggerController0 = o;

this.$ChatStickerTriggerController8.isMounted() && this.$ChatStickerTriggerController8.setProps({

triggeredWord: o

});

if (!this.$ChatStickerTriggerControllerd)

this.$ChatStickerTriggerController1.isMounted() && this.$ChatStickerTriggerController1.setProps({

stickers: n

});

};

m.prototype.$ChatStickerTriggerControllerb = function() {

clearTimeout(this.$ChatStickerTriggerControllera);

this.$ChatStickerTriggerController9 = false;

this.$ChatStickerTriggerControllerd = false;

this.$ChatStickerTriggerController0 = null;

this.$ChatStickerTriggerControllerc(null);

};

m.prototype.$ChatStickerTriggerControllerc = function(n) {

if (this.$ChatStickerTriggerController5 !== n) {

this.$ChatStickerTriggerController5 = n;

k.findTriggerForInput(this.$ChatStickerTriggerController5, this.$ChatStickerTriggerControllere.bind(this));

}

};

m.prototype.destroy = function() {

h.unmountComponentAtNode(this.$ChatStickerTriggerController2);

clearTimeout(this.$ChatStickerTriggerControllera);

};

e.exports = m;

}, null);

\_\_d("ChatFileUploaderMixin", ["SubscriptionsHandler", "React"], function(a, b, c, d, e, f, g, h) {

var i = {

propTypes: {

attachmentsShelf: h.PropTypes.object,

onFileUploaderMounted: h.PropTypes.func,

onUploadFinished: h.PropTypes.func,

onUpdateAttachmentsShelf: h.PropTypes.func,

onSubmit: h.PropTypes.func

},

setUpSubscriptionsHandler: function(j) {

this.\_uploader = j;

this.\_subscriptions = new g();

this.\_subscriptions.addSubscriptions(this.\_uploader.subscribe(['all-uploads-completed', 'upload-canceled'], this.\_handleUploadFinished), this.\_uploader.subscribe('dom-updated', this.\_handleUpdateAttachmentsShelf), this.\_uploader.subscribe('submit', this.\_handleSubmit));

this.props.onFileUploaderMounted(this.\_uploader);

},

componentWillUnmount: function() {

this.\_subscriptions && this.\_subscriptions.release();

if (!this.props.uploaderIsFromHigherLevel)

this.\_uploader && this.\_uploader.destroy();

},

isUploading: function() {

return this.\_uploader.isUploading();

},

getAttachments: function() {

return this.\_uploader.getAttachments();

},

getImageFiles: function() {

return this.\_uploader.getImageFiles();

},

getVideoFiles: function() {

return this.\_uploader.getVideoFiles();

},

getAudioFiles: function() {

return this.\_uploader.getAudioFiles();

},

getFiles: function() {

return this.\_uploader.getFiles();

},

removeAttachments: function() {

return this.\_uploader.removeAttachments();

},

\_handleUploadFinished: function() {

if (this.props.onUploadFinished)

this.props.onUploadFinished();

},

\_handleUpdateAttachmentsShelf: function() {

if (this.props.onUpdateAttachmentsShelf)

this.props.onUpdateAttachmentsShelf();

},

\_handleSubmit: function() {

if (this.props.onSubmit)

this.props.onSubmit();

}

};

e.exports = i;

}, null);

\_\_d("ChatFileUploader.react", ["ChatFileUploaderMixin", "InlineBlock.react", "MercuryConfig", "React", "cx", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

'use strict';

var m = j.createClass({

displayName: 'ChatFileUploader',

mixins: [g],

getDefaultProps: function() {

return {

uploaderIsFromHigherLevel: true

};

},

shouldComponentUpdate: function() {

return false;

},

componentDidMount: function() {

var n = this.props.initFileUploaderFn(this.refs.form.getDOMNode(), this.refs.input.getDOMNode(), this.refs.attachID.getDOMNode());

this.setUpSubscriptionsHandler(n);

},

render: function() {

return (j.createElement("form", {

ref: "form",

action: i.upload\_url,

method: "post"

}, j.createElement("input", {

type: "hidden",

ref: "attachID",

name: "attach\_id"

}), j.createElement(h, {

className: "\_m \_4q60 itemLabel"

}, j.createElement("input", {

type: "file",

className: "\_n",

name: "attachment[]",

multiple: "true",

ref: "input"

}), j.createElement("span", {

className: "\_4q61 itemAnchor"

}, "Dateien hinzuf\u00fcgen ..."))));

}

});

e.exports = m;

}, null);

\_\_d("ChatTabMenu.react", ["ChatFileUploader.react", "ChatPrivacyActionController", "ChatConfig", "MercuryConfig", "MercuryIDs", "MercuryFileUploader", "MercuryThreadMuter", "ReactComponentWithPureRenderMixin", "React", "PopoverMenu.react", "ReactMenu", "MenuSeparator.react", "Toggler", "TrackingNodes", "WebMessengerThreadPermalinks", "cx", "fbt", "goURI", "MercuryThreadActions"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x) {

'use strict';

var y = b('MercuryThreadActions').get(), z = q.Item, aa = o.createClass({

displayName: 'ChatTabMenu',

mixins: [n],

propTypes: {

onFileUploaderMounted: o.PropTypes.func,

updatePrivacyLinkFunc: o.PropTypes.func,

nameConversationFunc: o.PropTypes.func,

leaveConversationFunc: o.PropTypes.func,

showParticipantsFunc: o.PropTypes.func,

clearHistoryFunc: o.PropTypes.func,

reportSpamFunc: o.PropTypes.func,

createGroupFunc: o.PropTypes.func,

switchToWorkFunc: o.PropTypes.func,

isMultichat: o.PropTypes.bool,

isEmptyChat: o.PropTypes.bool,

hasSwitchToWork: o.PropTypes.bool,

showAddFriend: o.PropTypes.func,

threadID: o.PropTypes.string,

attachmentsShelf: o.PropTypes.object,

onUploadFinished: o.PropTypes.func,

onUpdateAttachmentsShelf: o.PropTypes.func,

onSubmit: o.PropTypes.func,

enableDesktopNotif: o.PropTypes.func

},

getInitialState: function() {

return {

url: null

};

},

getDefaultProps: function() {

return {

show: true

};

},

componentDidMount: function() {

u.getThreadURI(this.props.threadID, function(ba) {

this.isMounted() && this.setState({

url: ba

});

}.bind(this));

if (!this.props.isMultiChat)

this.\_privacyActionController = new h(k.getUserIDFromThreadID(this.props.threadID), this.props.updatePrivacyLinkFunc);

},

\_initFileUploader: function(ba, ca, da) {

if (this.\_fileUploader) {

this.\_fileUploader.updateElements(ba, ca, da);

} else

this.\_fileUploader = new l(this.props.attachmentsShelf, ba, ca, da);

return this.\_fileUploader;

},

componentWillUnmount: function() {

this.\_privacyActionController && this.\_privacyActionController.destroy();

this.\_fileUploader && this.\_fileUploader.destroy();

},

\_unmuteThread: function() {

y.unmute(this.props.threadID);

s.hide();

},

\_togglePrivacyFunc: function() {

this.\_privacyActionController.togglePrivacy();

},

getURL: function() {

return this.state.url;

},

\_renderFileUploader: function() {

return (o.createElement(z, {

className: "\_37\_x"

}, o.createElement(g, {

onFileUploaderMounted: this.props.onFileUploaderMounted,

attachmentsShelf: this.props.attachmentsShelf,

initFileUploaderFn: this.\_initFileUploader,

onUploadFinished: this.props.onUploadFinished,

onUpdateAttachmentsShelf: this.props.onUpdateAttachmentsShelf,

onSubmit: this.props.onSubmit

})));

},

\_renderOpenFullConversation: function() {

if (this.state.url)

return (o.createElement(z, {

onclick: function() {

return x(this.state.url);

}.bind(this)

}, "Vollst\u00e4ndige Unterhaltung anzeigen"));

return null;

},

\_renderAddFriend: function() {

if (!this.props.showAddFriend)

return null;

return (o.createElement(z, {

onclick: this.props.showAddFriend

}, "Freunde zum Chat hinzuf\u00fcgen ..."));

},

\_renderTogglePrivacy: function() {

if (!this.props.isMultichat)

return (o.createElement(z, {

onclick: this.\_togglePrivacyFunc

}, this.props.privacyText));

return null;

},

\_renderMuteConversation: function() {

if (!this.props.isMuted) {

var ba = m.showMuteChangeDialog.bind(null, this.props.threadID);

return (o.createElement(z, {

onclick: ba

}, "Unterhaltung stumm schalten ..."));

} else

return (o.createElement(z, {

onclick: this.\_unmuteThread

}, "Stummschaltung aufheben"));

},

\_renderClearConversation: function() {

if (!this.props.isMultichat)

return (o.createElement(z, {

onclick: this.props.clearHistoryFunc

}, "Fenster leeren"));

return null;

},

\_renderReportSpam: function() {

if (!this.props.isMultichat)

return (o.createElement(z, {

onclick: this.props.reportSpamFunc

}, "Als Spam oder Missbrauch melden ..."));

return null;

},

\_renderCreateGroup: function() {

if (this.props.isMultichat)

return (o.createElement(z, {

onclick: this.props.createGroupFunc

}, "Gruppe gr\u00fcnden"));

return null;

},

\_renderSwitchToWork: function() {

if (this.props.isMultichat && i.get('has\_switch\_to\_work') && this.props.hasSwitchToWork)

return (o.createElement(z, {

onclick: this.props.switchToWorkFunc

}, "Move to Work Account"));

},

\_renderConversationName: function() {

if (this.props.isMultichat && this.state.url)

return (o.createElement(z, {

onclick: this.props.nameConversationFunc

}, "Name der Unterhaltung bearbeiten"));

return null;

},

\_renderLeaveConversation: function() {

if (this.props.isMultichat)

return (o.createElement(z, {

onclick: this.props.leaveConversationFunc

}, "Unterhaltung verlassen ..."));

return null;

},

\_renderEnableDesktopNotifications: function() {

if (j.DesktopNotificationsGK && "Notification" in window && Notification.permission !== "granted")

return (o.createElement(z, {

onclick: this.props.enableDesktopNotif

}, "Enable Desktop Notifications"));

return null;

},

\_renderPeopleList: function() {

if (this.props.isMultichat && this.state.url && i.get('chat\_remove\_participants'))

return (o.createElement(z, {

onclick: this.props.showParticipantsFunc

}, "Teilnehmer bearbeiten ..."));

return null;

},

render: function() {

var ba = this.\_renderReportSpam(), ca = this.\_renderCreateGroup(), da = o.createElement(q, {

className: (("chatReact") + (!this.props.show ? ' ' + "hidden\_elem" : ''))

}, this.\_renderOpenFullConversation(), this.\_renderFileUploader(), this.\_renderAddFriend(), this.\_renderTogglePrivacy(), this.\_renderConversationName(), this.\_renderPeopleList(), o.createElement(r, null), this.\_renderSwitchToWork(), this.\_renderMuteConversation(), this.\_renderClearConversation(), this.\_renderLeaveConversation(), ba ? (o.createElement(r, null)) : null, ba, ca ? (o.createElement(r, null)) : null, ca, this.\_renderEnableDesktopNotifications()), ea = "Optionen", fa = t.getTrackingInfo(t.types.DROPDOWN\_BUTTON);

return (o.createElement(p, {

menu: da

}, o.createElement("a", {

'data-ft': fa,

'data-hover': "tooltip",

'aria-label': ea,

'data-tooltip-alignh': "center",

className: (("button") + (this.props.isEmptyChat ? ' ' + "hidden\_elem" : ''))

})));

}

});

e.exports = aa;

}, null);

\_\_d("ChatDateBreak.react", ["React", "ReactPropTypes", "Timestamp.react", "cx", "formatDate", "joinClasses"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = g.createClass({

displayName: 'ChatDateBreak',

propTypes: {

date: h.instanceOf(Date).isRequired

},

shouldComponentUpdate: function(n) {

return this.props.date.getTime() !== n.date.getTime();

},

render: function() {

var n = this.props.date, o = k(n, {

today: 'g:ia',

withinWeek: 'l g:ia',

thisYear: 'F jS, g:ia',

older: 'F j, Y g:i a'

}), p = Math.round(n.getTime() / 1000);

return (g.createElement("div", Object.assign({}, this.props, {

className: l(this.props.className, "\_5w-5")

}), g.createElement("div", {

className: "\_5w-6"

}, g.createElement(i, {

time: p,

verbose: o,

text: o

}))));

}

});

e.exports = m;

}, null);

\_\_d("ChatAuthorPhotoBlock.react", ["MercuryParticipants", "ReactPropTypes", "React", "cx", "formatDate", "joinClasses"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = 86400000, n = i.createClass({

displayName: 'ChatAuthorPhotoBlock',

propTypes: {

authorID: h.string.isRequired,

hideName: h.bool,

hidePhoto: h.bool,

timestamp: h.number

},

getInitialState: function() {

return {

author: {

id: null,

name: '',

href: '#',

image\_src: ''

}

};

},

componentDidMount: function() {

this.updateAuthor(this.props.authorID);

},

componentWillReceiveProps: function(o) {

if (o.authorID != this.state.author.id)

this.updateAuthor(o.authorID);

},

render: function() {

return (i.createElement("div", Object.assign({}, this.props, {

className: l(this.props.className, "\_5yt9")

}), this.renderAuthorName(), this.renderAuthorPhoto(), this.props.children));

},

renderAuthorName: function() {

if (!this.props.hideName)

return (i.createElement("div", {

className: "\_5ys-",

ref: "name"

}, this.state.author.name));

},

renderAuthorPhoto: function() {

if (!this.props.hidePhoto) {

var o = null;

if (this.props.timestamp) {

var p = (Date.now() - this.props.timestamp > m) ? 'M jS, g:ia': 'g:ia';

o = k(new Date(this.props.timestamp), p);

}

return (i.createElement("a", {

'aria-label': o,

className: "\_5ys\_",

'data-hover': "tooltip",

'data-tooltip-position': "left",

href: this.state.author.href,

ref: "link"

}, i.createElement("img", {

src: this.state.author.image\_src,

ref: "image"

})));

}

},

updateAuthor: function(o) {

g.get(o, function(p) {

this.isMounted() && this.setState({

author: p

});

}.bind(this));

}

});

e.exports = n;

}, null);

\_\_d("ChatBubble.react", ["DOMDimensions", "MercuryMessageBody.react", "ReactPropTypes", "React", "cx", "joinClasses"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = null, n = j.createClass({

displayName: 'ChatBubble',

propTypes: {

attachments: i.node,

body: i.string,

maxWidth: i.number.isRequired

},

getInitialState: function() {

return {

bubbleWidth: this.props.maxWidth

};

},

componentDidMount: function() {

this.shrinkBubble();

},

componentWillReceiveProps: function(o) {

if (o.body != this.props.body)

this.setState({

bubbleWidth: this.props.maxWidth

});

},

componentDidUpdate: function(o, p) {

if (o.body != this.props.body)

this.shrinkBubble();

},

render: function() {

return (j.createElement("div", Object.assign({}, this.props, {

className: l(this.props.className, "\_5w1r"),

ref: "bubble",

style: Object.assign({}, (this.props.style || {}), {

maxWidth: this.state.bubbleWidth

})

}), j.createElement("div", null, j.createElement("span", {

className: "\_5yl5",

ref: "content"

}, j.createElement(h, {

body: this.props.body,

ranges: this.props.ranges

}))), this.props.attachments));

},

shrinkBubble: function() {

var o = this.refs.bubble.getDOMNode(), p = this.refs.content.getDOMNode(), q = g.getElementDimensions(o).width, r = g.getElementDimensions(p).width;

if (this.contentWraps() && r > 0 && r < this.props.maxWidth && q - r - this.getBoxWidth() > 0 && r != this.state.bubbleWidth)

this.setState({

bubbleWidth: r

});

},

contentWraps: function() {

var o = this.refs.content.getDOMNode(), p = o.getClientRects(), q = g.getElementDimensions(o).height;

return (p && p.length > 1 && p[0].height < q);

},

getBoxWidth: function() {

if (!m)

m = g.measureElementBox(this.refs.bubble.getDOMNode(), 'width', true, true, false);

return m;

}

});

e.exports = n;

}, null);

\_\_d("XStickerAssetControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/asset\/", {

sticker\_id: {

type: "Int",

required: true

},

image\_type: {

type: "Enum"

}

});

}, null);

\_\_d("XLinkshimLogControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/si\/ajax\/l\/render\_linkshim\_log\/", {

u: {

type: "String",

required: true

},

h: {

type: "String",

required: true

},

render\_verification: {

type: "Bool"

},

enc: {

type: "String"

},

d: {

type: "String"

}

});

}, null);

\_\_d("MercuryAttachmentRenderer", ["MercuryAttachmentTemplates", "MercuryAttachmentAudioClip.react", "Bootloader", "ChatAnimatedGifs", "SyncRequest.react", "CSS", "MercuryConstants", "DOM", "Event", "MercuryAttachment", "MercuryAttachmentType", "MercuryAttachmentVideo.react", "MercuryIDs", "MercuryParticipants", "MercuryViewer", "ProgressBar", "React", "Sticker.react", "StickerAssetType", "Style", "URI", "UserAgent\_DEPRECATED", "cx", "endsWith", "invariant", "tx", "updatePhotoProgressBar", "XStickerAssetControllerURIBuilder", "OrionMercuryAttachment", "MercuryMessages"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, aa, ba, ca, da, ea, fa, ga, ha) {

'use strict';

var ia = b('OrionMercuryAttachment').module, ja = b('MercuryMessages').get(), ka = ba.ie() <= 8;

function la(oa, pa) {

var qa = g[pa].build().setNodeContent('filename', oa.name), ra = qa.getNode('link');

ra.setAttribute('href', oa.url);

oa.rel && ra.setAttribute('rel', oa.rel);

l.addClass(qa.getRoot(), p.getAttachIconClass(oa.icon\_type));

return qa;

}

function ma(oa, pa) {

var qa = g[pa].build().setNodeContent('filename', oa.name);

l.addClass(qa.getRoot(), p.getAttachIconClass(oa.icon\_type));

return qa;

}

var na = {

renderAttachment: function(oa, pa, qa, ra, sa, ta) {

var ua = 100, va = oa ? 160: 400, wa = null, xa = null, ya = m.MercurySupportedShareType;

if (this.isErrorAttachment(pa))

wa = na.renderError(pa);

if (this.isShareAttachment(pa) && pa.share\_xhp)

if (pa.share\_data\_type === ya.FB\_BROWSE\_QUERY ||!oa)

xa = na.renderShareXHP(pa, qa.id);

if (this.isStickerAttachment(pa))

xa = na.renderSticker(pa, qa, oa, ta);

if (!xa && this.isShareAttachment(pa)) {

var za = pa.share\_data\_type;

switch (za) {

case ya.FB\_PHOTO:

xa = na.renderPreview(pa, qa, ra, sa);

break;

case ya.FB\_VIDEO:

xa = na.renderVideoThumb(pa);

break;

case ya.FB\_MUSIC\_ALBUM:

case ya.FB\_SONG:

case ya.FB\_PLAYLIST:

case ya.FB\_MUSICIAN:

case ya.FB\_RADIO\_STATION:

xa = na.renderMusic(pa);

break;

case ya.EXTERNAL:

case ya.FB\_TEMPLATE:

case ya.FB\_SOCIAL\_REPORT\_PHOTO:

xa = na.renderExternalLink(pa);

break;

case ya.FB\_COUPON:

case ya.FB\_EVENT:

case ya.FB\_SHOERACK\_INVITATION:

xa = na.renderChatXHP(pa);

break;

case ya.FB\_SOCIAL\_RESOLUTION:

case ya.FB\_STATUS:

xa = na.renderSocialResolution(pa);

break;

case ya.FB\_SYNC\_REQUEST:

xa = na.renderSyncRequest(pa);

break;

case ya.FB\_OPEN\_GRAPH:

xa = na.renderOpenGraph(pa);

break;

case ya.FB\_ORION:

if (ia)

xa = na.renderOrionMercuryAttachment(pa);

break;

default:

if (pa.name)

xa = na.renderShareLink(pa, qa && qa.id, oa);

break;

}

}

if (!xa && pa.preview\_loading)

xa = na.renderPreview(pa, qa, ra, sa);

if (!xa && this.isVideoAttachment(pa)) {

xa = n.create('div');

w.render(na.renderVideo(pa, oa), xa);

}

if (!xa && pa.preview\_url)

xa = na.renderPreview(pa, qa, ra, sa);

if (!xa && this.isFileAttachment(pa))

if (pa.metadata && pa.metadata.type == m.MercuryAttachmentAudioClip) {

xa = n.create('div');

var ab = na.renderAudioClip(pa, qa.message\_id, ua, va);

w.render(ab, xa);

} else

xa = oa ? na.renderFileLink(pa) : na.renderExtendedFileLink(pa);

return {

error: wa,

content: xa,

bubblePreferred: this.isBubblePreferred(pa)

};

},

isBubblePreferred: function(oa) {

return !this.isStickerAttachment(oa)&&!this.isSyncRequestAttachment(oa);

},

renderError: function(oa) {

var pa = g[':fb:mercury:attachment:error'].build();

n.appendContent(pa.getNode('error'), oa.error\_msg);

return pa.getRoot();

},

renderSocialResolution: function(oa) {

var pa = g[':fb:mercury:attachment:social-resolution'].build();

pa.setNodeContent('post', oa.share\_xhp);

return pa.getRoot();

},

renderChatXHP: function(oa) {

var pa = g[':fb:mercury:attachment:social-resolution'].build();

pa.setNodeContent('post', oa.chat\_xhp);

return pa.getRoot();

},

renderOpenGraph: function(oa) {

var pa = oa.share\_xhp.cloneNode(true);

if (ba.firefox())

z.set(pa, 'minWidth', '180px');

return pa;

},

renderExternalLink: function(oa) {

var pa = g[':fb:mercury:attachment:external-link'].build().setNodeContent('name', oa.name);

oa.base\_url && pa.setNodeContent('shortLink', oa.base\_url);

var qa = pa.getNode('preview'), ra = pa.getNode('image-link');

ra.setAttribute('href', oa.url);

oa.rel && ra.setAttribute('rel', oa.rel);

if (oa.preview\_url) {

var sa = pa.getNode('preview-image'), ta = oa.preview\_url, ua = aa(j.getRawUrlFromSafeUrl(ta));

if (da(ua.getPath(), '.gif')) {

ta = aa('/animated.php').setSecure(true).setDomain('www.fbsbx.com').addQueryData('url', ua.toString());

l.addClass(qa, "\_dri");

if (oa.name && oa.name.\_\_html == ua.toString())

pa.setNodeContent('name', '');

}

sa.setAttribute('src', ta);

l.addClass(qa, oa.preview\_class);

l.show(sa);

} else {

l.addClass(pa.getRoot(), 'noMedia');

l.hide(qa);

}

pa.getNode('name').setAttribute('href', oa.url);

d(['LinkshimHandler'], function(va) {

va.setUpLinkshimHandling(pa.getNode('name'));

va.setUpLinkshimHandling(pa.getNode('image-link'));

});

if (oa.rel)

pa.getNode('name').setAttribute('rel', oa.rel);

return pa.getRoot();

},

renderFileLink: function(oa) {

var pa = null;

if (oa.url === '') {

pa = ':fb:mercury:attachment:file-name';

return ma(oa, pa).getRoot();

} else {

pa = ':fb:mercury:attachment:file-link';

return la(oa, pa).getRoot();

}

},

renderAudioClip: function(oa, pa, qa, ra) {

var sa = oa.metadata.duration / 1000, ta = 200;

if (qa && ra)

if (sa < 5) {

ta = qa;

} else

ta = (1 - Math.pow(10, (sa - 5)/-30)) \* (ra - qa) + qa;

return w.createElement(h, {

src: oa.url,

duration: oa.metadata.duration / 1000,

showHelp: false,

width: ta

});

},

renderVideo: function(oa, pa) {

var qa = {

height: oa.preview\_height,

width: oa.preview\_width

}, ra = pa ? {

width: 160,

height: 120

}

: {

width: 266,

height: 200

};

return w.createElement(r, {

duration: oa.metadata.duration,

name: oa.name,

pageID: oa.metadata.pageid,

size: ra,

thumbnailSize: qa,

thumbnail: oa.thumbnail\_url,

videoID: oa.metadata.fbid

});

},

renderExtendedFileLink: function(oa) {

var pa = null;

if (oa.url === '') {

pa = ':fb:mercury:attachment:file-name';

return ma(oa, pa).getRoot();

}

pa = ':fb:mercury:attachment:extended-file-link';

var qa = la(oa, pa);

if (oa.open\_url) {

var ra = qa.getNode('openLinkContainer');

l.show(ra);

var sa = qa.getNode('openFile');

sa.setAttribute('href', oa.open\_url);

}

var ta = qa.getNode('downloadFile');

ta.setAttribute('href', oa.url);

oa.rel && ta.setAttribute('rel', oa.rel);

return qa.getRoot();

},

renderMusic: function(oa) {

var pa = g[':fb:mercury:attachment:music'].build().setNodeContent('filename', oa.name), qa = pa.getNode('link');

qa.setAttribute('href', oa.url);

qa.setAttribute('target', '\_blank');

oa.rel && qa.setAttribute('rel', oa.rel);

var ra = pa.getNode('image-link');

ra.setAttribute('href', oa.url);

oa.rel && ra.setAttribute('rel', oa.rel);

var sa = pa.getNode('preview-image');

sa.setAttribute('src', oa.preview\_url);

l.show(sa);

l.addClass(pa.getNode('icon\_link'), 'MercuryMusicIcon');

return pa.getRoot();

},

renderSyncRequest: function(oa) {

var pa = k(oa.metadata), qa = n.create('div');

w.render(pa, qa);

return qa;

},

renderOrionMercuryAttachment: function(oa) {

var pa = w.createElement(ia, Object.assign({}, oa.metadata)), qa = n.create('div');

w.render(pa, qa);

return qa;

},

resizeContain: function(oa, pa) {

var qa = oa.width / oa.height, ra = pa.width / pa.height;

if (ra < qa) {

return {

width: Math.min(oa.height \* ra, pa.width),

height: Math.min(oa.height, pa.height)

};

} else

return {

width: Math.min(oa.width, pa.width),

height: Math.min(oa.width / ra, pa.height)

};

},

renderPreview: function(oa, pa, qa, ra) {

var sa = g[':fb:mercury:attachment:preview'].build(), ta = sa.getNode('image-link');

if (oa) {

if (oa.url) {

var ua = new aa(oa.url).getQueryData().uri;

if (ua && oa.rel === 'async') {

ta.setAttribute('href', ua);

ta.setAttribute('ajaxify', oa.url);

} else

ta.setAttribute('href', oa.url);

}

oa.rel && ta.setAttribute('rel', oa.rel);

var va;

if (oa.preview\_uploading) {

var wa = sa.getNode('cancel-button-container');

l.show(wa);

var xa = sa.getNode('cancel-button'), ya = o.listen(xa, 'click', function() {

oa.upload\_canceled(oa.upload\_id);

l.hide(sa.getRoot());

ya.remove();

});

oa.on\_success(function(event, gb) {

if (gb.upload\_id == oa.upload\_id) {

l.hide(wa);

ya.remove();

}

});

var za = sa.getNode('progress-bar'), ab = new v(za), bb = sa.getNode('progress-bar-container');

ea(oa.upload\_id);

oa.on\_progress(function(event, gb) {

if (gb.upload\_id == oa.upload\_id) {

l.removeClass(bb, "\_395w");

l.show(bb);

ga(ab, gb.event);

}

});

if (oa.on\_resizing\_progress)

oa.on\_resizing\_progress(function(event, gb) {

if (gb.upload\_id == oa.upload\_id) {

l.addClass(bb, "\_395w");

l.show(bb);

ab.setPosition(100 \* gb.event.written / gb.event.total);

}

});

l.addClass(ta, "\_57jm");

if (qa >= 176) {

va = '/images/photos/dots\_large.png';

} else if (qa >= 86) {

va = '/images/photos/dots\_medium.png';

} else

va = '/images/photos/dots\_small.png';

z.set(ta, 'width', qa + 'px');

z.set(ta, 'max-width', qa + 'px');

if (oa.preview\_width && oa.preview\_height)

z.set(ta, 'padding-bottom', ((oa.preview\_height / oa.preview\_width) \* 100) + '%');

} else if (oa.preview\_loading) {

l.addClass(ta, "\_5xdv");

if (ra === 'contain' && oa.preview\_width && oa.preview\_height) {

z.set(ta, 'width', oa.preview\_width + 'px');

z.set(ta, 'height', oa.preview\_height + 'px');

}

if (ra === 'cover'&&!ka)

l.addClass(ta, "\_55pj");

} else if (oa.metadata && oa.metadata.fbid) {

va = aa('/ajax/mercury/attachments/photo.php').addQueryData({

fbid: oa.metadata.fbid,

request\_user\_id: oa.metadata.pageid,

mode: ra,

width: qa,

height: qa

}).toString();

var cb = ta.getAttribute('ajaxify');

ta.removeAttribute('ajaxify');

ta.removeAttribute('rel');

o.listen(ta, 'click', function(event) {

event.prevent();

i.loadModules(["MessagesViewer"], function(gb) {

gb.bootstrapWithConfig({

src: va,

endpoint: cb,

fbid: oa.metadata.fbid,

dimensions: oa.metadata.dimensions,

disablePaging: pa && pa.attachments.length == 1

}, ta);

});

});

} else {

var db = aa(oa.preview\_url);

ra && db.addQueryData({

mode: ra

});

qa && db.addQueryData({

width: qa,

height: qa

});

va = db.toString();

}

var eb = sa.getNode('preview-image');

if (va) {

if (ra === 'contain' && oa.preview\_width && oa.preview\_height) {

var fb = na.resizeContain({

width: qa,

height: qa

}, {

width: oa.preview\_width,

height: oa.preview\_height

});

eb.setAttribute('width', fb.width);

eb.setAttribute('height', fb.height);

}

if (oa.preview\_uploading || (ra === 'cover'&&!ka)) {

l.addClass(ta, "\_55pj");

z.set(ta, 'backgroundImage', 'url(' + va + ')');

} else {

eb.onload = function() {

eb.removeAttribute('width');

eb.removeAttribute('height');

};

eb.setAttribute('src', va);

}

}

if (pa)

this.renderReportRespondLink(sa.getRoot(), oa, pa.message\_id);

}

return sa.getRoot();

},

renderShareLink: function(oa, pa, qa) {

var ra = g[':fb:mercury:attachment:share-link'].build().setNodeContent('name', oa.name), sa = ra.getNode('link');

sa.setAttribute('href', oa.url);

oa.rel && sa.setAttribute('rel', oa.rel);

return ra.getRoot();

},

renderVideoThumb: function(oa) {

var pa = g[':fb:mercury:attachment:video-thumb'].build(), qa = pa.getNode('thumb');

qa.setAttribute('href', oa.url);

qa.setAttribute('rel', oa.rel);

var ra = n.find(pa.getRoot(), 'img');

ra.src = oa.preview\_url;

return pa.getRoot();

},

renderShareXHP: function(oa, pa) {

var qa = n.create('div');

if (oa) {

n.appendContent(qa, oa.share\_xhp);

this.renderReportRespondLink(qa, oa, pa);

}

return qa;

},

renderSticker: function(oa, pa, qa, ra) {

var sa = qa ? 'chatScrolled/': 'messengerScrolled/';

sa += pa.thread\_id;

var ta = n.create('div');

l.addClass(ta, 'stickerContainer');

var ua = null;

if (oa.metadata.stickerID)

ua = oa.metadata.stickerID.toString();

var va = null;

if (oa.metadata.packID)

va = oa.metadata.packID.toString();

var wa = new ha().setInt('sticker\_id', ua), xa = null, ya = null;

if (window.devicePixelRatio && window.devicePixelRatio > 1) {

ya = oa.metadata.paddedSpriteURI2x;

xa = oa.metadata.spriteURI2x;

} else {

ya = oa.metadata.paddedSpriteURI;

xa = oa.metadata.spriteURI;

}

var za = w.createElement(x, {

animationTrigger: "hover",

className: "mvs",

frameCount: oa.metadata.frameCount || 1,

frameRate: oa.metadata.frameRate || 83,

framesPerCol: oa.metadata.framesPerCol || 1,

framesPerRow: oa.metadata.framesPerRow || 1,

onStickerClick: ra,

packID: va,

paddedSpriteURI: ya,

sourceHeight: oa.metadata.height,

sourceURI: wa.setEnum('image\_type', y.IMAGE).getURI().toString(),

sourceWidth: oa.metadata.width,

spriteURI: xa,

stickerID: ua,

subscribedThreadID: sa

});

w.render(za, ta);

return ta;

},

renderReportRespondLink: function(oa, pa, qa) {

if (!pa.is\_social\_report\_attachment)

return null;

switch (pa.share\_data\_type) {

case m.MercurySupportedShareType.FB\_PHOTO:

break;

case m.MercurySupportedShareType.FB\_SOCIAL\_REPORT\_PHOTO:

return null;

default:

return null;

}

var ra = null;

if (qa)

ra = ja.getMessagesFromIDs([qa])[0];

if (!ra)

return null;

if (ra.author === u.getID())

return null;

var sa = null;

t.get(ra.author, function(ta) {

sa = n.create('a', {

rel: 'dialog-post',

className: "\_z6l",

id: 'respond-link',

ajaxify: aa('/ajax/report/social\_resolution/post/').setQueryData({

attachment\_fbid: pa.attach\_id,

post\_fbid: pa.shared\_object\_id,

sender\_id: s.getUserIDFromParticipantID(ta.id)

}).toString()

});

n.setContent(sa, fa.\_("Antworte auf {name}s Anfrage", {

name: ta.name

}));

n.appendContent(oa, sa);

});

},

renderPhotoAttachments: function(oa, pa, qa, ra) {

var sa = oa.length;

if (!sa)

return null;

var ta = n.create('div', {

className: "\_55pk"

});

if (sa === 1) {

var ua = na.renderPreview(oa[0], pa, qa, 'contain');

n.appendContent(ta, ua);

return ta;

}

var va = (sa == 2 || sa == 4) ? 2: 3, wa = (qa - (va - 1) \* ra) / va, xa = Math.ceil(sa / va), ya = xa \* wa + (xa - 1) \* ra, za = n.create('div', {

className: "\_55pm",

style: 'padding-bottom: ' + (ya / qa \* 100) + '%;'

});

n.appendContent(ta, za);

for (var ab = 0; ab < sa; ++ab) {

var bb = na.renderPreview(oa[ab], pa, wa, 'cover'), cb = ab%va, db = Math.floor(ab / va);

l.addClass(bb, "\_55pn");

z.apply(bb, {

width: (wa / qa \* 100) + '%',

left: ((cb \* (wa + ra)) / qa \* 100) + '%',

top: ((db \* (wa + ra)) / ya \* 100) + '%'

});

n.appendContent(za, bb);

}

return ta;

},

isPhotoAttachment: function(oa) {

return oa.attach\_type == q.PHOTO || (oa.attach\_type == q.FILE && oa.preview\_url);

},

isVideoAttachment: function(oa) {

return oa.attach\_type == q.VIDEO;

},

isShareAttachment: function(oa) {

return oa.attach\_type == q.SHARE;

},

isFileAttachment: function(oa) {

return oa.attach\_type == q.FILE;

},

isErrorAttachment: function(oa) {

return oa.attach\_type == q.ERROR;

},

isStickerAttachment: function(oa) {

return oa.attach\_type == q.STICKER;

},

isSyncRequestAttachment: function(oa) {

var pa = m.MercurySupportedShareType;

return this.isShareAttachment(oa) && oa.share\_data\_type == pa.FB\_SYNC\_REQUEST;

},

sortAttachmentsStablyByType: function(oa) {

var pa = [na.isPhotoAttachment, na.isShareAttachment, na.isFileAttachment, na.isErrorAttachment];

pa.push(function(ra) {

return true;

});

var qa = pa.map(function(ra) {

return [];

});

oa.forEach(function(ra) {

for (var sa = 0; sa < pa.length; sa++)

if (pa[sa](ra)) {

qa[sa].push(ra);

break;

}

});

return Array.prototype.concat.apply([], qa);

}

};

e.exports = na;

}, null);

\_\_d("MercuryAttachments.react", ["MercuryAttachmentRenderer", "DOM", "Event", "ImmutableObject", "React", "SubscriptionsHandler", "cx", "emptyFunction", "joinClasses"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o) {

'use strict';

var p = 176, q = 4, r = k.createClass({

displayName: 'MercuryAttachments',

propTypes: {

message: k.PropTypes.instanceOf(j).isRequired,

attachments: k.PropTypes.array.isRequired,

isChat: k.PropTypes.bool,

onImageLoad: k.PropTypes.func,

onStickerClick: k.PropTypes.func

},

getDefaultProps: function() {

return {

isChat: false,

onImageLoad: n,

onStickerClick: n

};

},

componentDidMount: function() {

this.\_subscriptionsHandler = new l();

this.renderAttachments();

},

componentDidUpdate: function() {

this.renderAttachments();

},

componentWillUnmount: function() {

this.\_subscriptionsHandler.release();

},

render: function() {

var s = this.props.attachments.filter(g.isPhotoAttachment).length > 0;

return (k.createElement("div", Object.assign({}, this.props, {

className: o(this.props.className, (("\_5h9y") + (s ? ' ' + "\_zl6" : '')))

})));

},

renderAttachments: function() {

var s = this.getDOMNode();

h.empty(s);

var t = this.props.message, u = g.sortAttachmentsStablyByType(this.props.attachments), v = g.renderPhotoAttachments(u.filter(g.isPhotoAttachment), t, p, q);

v && h.appendContent(s, v);

for (var w = 0; w < u.length; w++) {

var x = u[w];

if (g.isPhotoAttachment(x))

continue;

var y = g.renderAttachment(this.props.isChat, x, t, null, null, this.props.onStickerClick);

y.error && h.appendContent(s, y.error);

y.content && h.appendContent(s, y.content);

}

h.scry(s, 'img').forEach(function(z) {

this.\_subscriptionsHandler.addSubscriptions(i.listen(z, 'load', function() {

this.props.onImageLoad(z);

}.bind(this)));

}.bind(this));

}

});

e.exports = r;

}, null);

\_\_d("MercurySpoofWarning.react", ["MercuryParticipants", "ReactPropTypes", "React", "fbt"], function(a, b, c, d, e, f, g, h, i, j) {

var k = i.createClass({

displayName: 'MercurySpoofWarning',

propTypes: {

authorID: h.string.isRequired

},

getInitialState: function() {

return {

author: {

name: ''

}

};

},

componentWillMount: function() {

this.componentWillReceiveProps(this.props);

},

componentWillReceiveProps: function(l) {

g.get(l.authorID, function(m) {

return this.setState({

author: m

});

}.bind(this));

},

render: function() {

return (i.createElement("div", Object.assign({}, this.props), j.\_("{name\_or\_email} kann nicht als Absender best\u00e4tigt werden.", [j.param("name\_or\_email", this.state.author.name)])));

}

});

e.exports = k;

}, null);

\_\_d("ChatMessage.react", ["ChatAnimatedGifs", "ChatAuthorPhotoBlock.react", "ChatBubble.react", "ChatConfig", "CSS", "ImmutableObject", "MercuryAttachment", "MercuryAttachments.react", "MercuryAttachmentRenderer", "MercuryErrorInfo", "MercuryIDs", "MercuryMessageError.react", "MercurySpoofWarning.react", "React", "ReactPropTypes", "cx", "fbt", "formatDate", "isRTL", "joinClasses", "MercuryMessages"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z) {

'use strict';

var aa = b('MercuryMessages').get(), ba = t.createClass({

displayName: 'ChatMessage',

propTypes: {

message: u.instanceOf(l).isRequired,

maxBubbleWidth: u.number.isRequired,

onImageLoad: u.func,

onStickerClick: u.func

},

componentDidMount: function() {

k.conditionClass(this.getDOMNode(), "\_2cnu", this.isOneLine());

},

isOneLine: function() {

var ca = this.props.message;

return aa.isInbound(ca)&&!this.refs.attachmentsInside && this.refs.bubble&&!this.refs.bubble.contentWraps();

},

shouldComponentUpdate: function(ca, da) {

return ca.message !== this.props.message;

},

render: function() {

var ca = this.props.message, da = aa.isInbound(ca), ea = q.isMultichat(ca.thread\_id), fa = y(ca.body), ga = j.get('chat\_fading\_bubbles')&&!da && aa.isSending(ca), ha = p.hasErrorStatus(ca), ia = (("\_5wd4") + (da ? ' ' + "\_1nc7" : '') + (!da ? ' ' + "\_1nc6" : '') + (da && ea ? ' ' + "\_5ysy" : '') + (ga ? ' ' + "\_4oe5" : '') + (ha ? ' ' + "\_1zcs" : '') + (fa ? ' ' + "direction\_rtl" : '') + (!fa ? ' ' + "direction\_ltr" : '')), ja = new Date(ca.timestamp);

return (t.createElement(h, Object.assign({}, this.props, {

authorID: ca.author,

className: z(this.props.className, ia),

hideName: !da ||!ea,

hidePhoto: !da,

timestamp: ca.timestamp,

title: da ? null: x(ja, 'g:ia')

}), this.\_renderInner(), this.\_renderUndertext()));

},

\_renderInner: function() {

if (this.props.message.is\_filtered\_content)

return (t.createElement("div", {

className: "\_5wd9",

ref: "inner"

}, this.\_renderFilteredContent()));

return (t.createElement("div", {

className: "\_5wd9",

ref: "inner"

}, this.\_renderBubble(), this.\_renderAttachments('outside')));

},

\_renderUndertext: function() {

if (this.props.message.is\_filtered\_content)

return null;

return (t.createElement("div", {

className: "\_3ry4",

ref: "undertext"

}, t.createElement(r, {

className: "\_5wda clearfix",

message: this.props.message,

ref: "status"

}), this.\_renderSpoofWarning()));

},

\_renderSpoofWarning: function() {

return (this.props.message.is\_spoof\_warning ? t.createElement(s, {

authorID: this.props.message.author,

className: "\_5wdb"

}) : null);

},

\_renderFilteredContent: function() {

return t.createElement("div", {

className: "\_5wdc uiBoxYellow clearfix"

}, "Die Nachricht ist nicht mehr verf\u00fcgbar, da sie als missbr\u00e4uchlich bzw. Spam markiert wurde.");

},

\_renderAttachments: function(ca) {

var da = this.props.message, ea = ca == 'inside' ? 'attachmentsInside': 'attachmentsOutside', fa = m.get(da).filter(function(ga) {

var ha = o.isBubblePreferred(ga);

return (ha && ca === 'inside') || (!ha && ca === 'outside');

});

if (fa.length > 0)

return t.createElement(n, {

className: "\_5wdd clearfix",

isChat: true,

message: da,

attachments: fa,

ref: ea,

onImageLoad: this.props.onImageLoad,

onStickerClick: this.props.onStickerClick

});

},

\_renderBubble: function() {

var ca = this.props.message, da = ca.body, ea = this.\_renderAttachments('inside');

if (g.shouldHideBody(ca))

da = '';

if (da || ea)

return (t.createElement("div", {

className: "\_5wde"

}, t.createElement(i, {

attachments: ea,

body: da,

className: "\_5wdf",

maxWidth: this.props.maxBubbleWidth,

ranges: ca.ranges,

ref: "bubble"

})));

}

});

e.exports = ba;

}, null);

\_\_d("ChatVideoCallLink.react", ["Arbiter", "ChatVisibility", "MercuryViewer", "PresencePrivacy", "React", "ReactPropTypes", "SubscriptionsHandler", "VideoCallCore", "cx", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

var q = k.createClass({

displayName: 'ChatVideoCallLink',

propTypes: {

message: l.object.isRequired,

onClick: l.func.isRequired

},

componentDidMount: function() {

this.\_subscriptions = new m();

this.\_subscriptions.addSubscriptions(g.subscribe(['buddylist/availability-changed'], this.availabilityChanged), j.subscribe(['privacy-changed', 'privacy-availability-changed'], this.availabilityChanged));

},

componentWillUnmount: function() {

this.\_subscriptions.release();

},

render: function() {

return (k.createElement("a", {

className: (("callBackLink") + (this.shouldHideLink() ? ' ' + "hidden\_elem" : '')),

'data-gt': JSON.stringify({

videochat: 'clicked\_callback\_link'

}),

href: "#",

onClick: this.linkClicked

}, this.renderLinkText()));

},

renderLinkText: function() {

if (this.props.message.log\_message\_data.event\_name == 'install\_canceled') {

return ("Installation erneut versuchen und zur\u00fcckrufen.");

} else if (!this.props.message.log\_message\_data.event\_name && this.props.message.log\_message\_data.callee == i.getID()&&!this.props.message.log\_message\_data.answered)

return ("Rufe zur\u00fcck.");

},

shouldHideLink: function() {

return !h.isOnline() ||!n.availableForCall(this.props.message.log\_message\_data.to);

},

linkClicked: function() {

var r;

if (this.props.message.log\_message\_data.event\_name == 'install\_canceled') {

r = 'callback\_cancelinstall\_link';

} else if (!this.props.message.log\_message\_data.event\_name && this.props.message.log\_message\_data.callee == i.getID()&&!this.props.message.log\_message\_data.answered)

r = 'callback\_link';

this.props.onClick(this.props.message.log\_message\_data.to, this.props.message.thread\_id, r);

},

availabilityChanged: function() {

this.forceUpdate();

}

});

e.exports = q;

}, null);

\_\_d("MercuryLogMessageRenderer", ["MercuryAttachmentRenderer", "CSS", "DOM", "HovercardLink", "MercuryViewer", "MercuryLogMessageType", "MercuryParticipants", "React", "Image.react", "TooltipLink.react", "cx", "fbt", "ix", "OrionMercuryLogMessage"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s) {

var t = b('OrionMercuryLogMessage').module, u = {

renderText: function(oa, pa) {

switch (oa.log\_message\_type) {

case l.SUBSCRIBE:

w(oa, pa);

break;

case l.UNSUBSCRIBE:

ba(oa, pa);

break;

case l.VIDEO\_CALL:

ca(oa, pa);

break;

case l.PHONE\_CALL:

ea(oa, pa);

break;

case l.THREAD\_NAME:

fa(oa, pa);

break;

case l.THREAD\_IMAGE:

ga(oa, pa);

break;

case l.WALLPAPER:

ha(oa, pa);

break;

case l.SERVER\_ERROR:

ia(oa, pa);

break;

case l.ORION:

if (t)

ja(oa, pa);

break;

}

},

renderIcon: function(oa) {

return (n.createElement(o, {

className: v(oa),

src: s('images/spacer.gif')

}));

},

renderLegacy: function(oa, pa, qa, ra) {

v(ra).split(' ').forEach(function(sa) {

return sa && h.addClass(oa, sa);

});

this.renderText(ra, function(sa) {

n.render(n.createElement("span", null, sa), pa);

});

this.renderAttachmentLegacy(qa, ra);

},

renderAttachmentLegacy: function(oa, pa) {

if (pa.log\_message\_type == l.THREAD\_IMAGE) {

var qa = pa.log\_message\_data.image;

if (qa) {

var ra = g.renderPreview(qa.preview\_url ? qa : null);

i.setContent(oa, ra);

h.addClass(ra, "\_z6a");

h.show(oa);

}

}

}

};

e.exports = u;

function v(oa) {

var pa = oa.log\_message\_type, qa = oa.log\_message\_data;

return (("\_5wzu") + (pa == l.SUBSCRIBE ? ' ' + "\_5wzj" : '') + (pa == l.UNSUBSCRIBE ? ' ' + "\_5wzk" : '') + (pa == l.THREAD\_NAME ? ' ' + "\_5wzl" : '') + (pa == l.THREAD\_IMAGE ? ' ' + "\_5wzm" : '') + (pa == l.VIDEO\_CALL && (qa.answered || na(oa)) ? ' ' + "\_5wzn" : '') + (pa == l.VIDEO\_CALL&&!(qa.answered || na(oa)) ? ' ' + "\_5wzo" : '') + (pa == l.PHONE\_CALL && qa.answered ? ' ' + "\_5wzp" : '') + (pa == l.PHONE\_CALL&&!qa.answered ? ' ' + "\_5wzq" : '') + (pa == l.SERVER\_ERROR ? ' ' + "\_5wzr" : ''));

}

function w(oa, pa) {

var qa = ma(oa.log\_message\_data.added\_participants);

switch (qa.length) {

case 1:

x(oa, pa, qa);

break;

case 2:

y(oa, pa, qa);

break;

case 3:

z(oa, pa, qa);

break;

default:

aa(oa, pa, qa);

break;

}

}

function x(oa, pa, qa) {

var ra = [oa.author, qa[0]];

m.getMulti(ra, function(sa) {

if (oa.author == k.getID()) {

pa(r.\_("Du hast {subscriber1} hinzugef\u00fcgt.", [r.param("subscriber1", la(sa[qa[0]]))]));

} else if (qa[0] == k.getID()) {

pa(r.\_("{actor} hat dich hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author]))]));

} else

pa(r.\_("{actor} hat {subscriber1} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber1", la(sa[qa[0]]))]));

});

}

function y(oa, pa, qa) {

var ra = [oa.author].concat(qa);

m.getMulti(ra, function(sa) {

if (oa.author == k.getID()) {

pa(r.\_("Du hast {subscriber1} und {subscriber2} hinzugef\u00fcgt.", [r.param("subscriber1", la(sa[qa[0]])), r.param("subscriber2", la(sa[qa[1]]))]));

} else if (qa[0] == k.getID()) {

pa(r.\_("{actor} hat dich und {subscriber2} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber2", la(sa[qa[1]]))]));

} else

pa(r.\_("{actor} hat {subscriber1} und {subscriber2} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber1", la(sa[qa[0]])), r.param("subscriber2", la(sa[qa[1]]))]));

});

}

function z(oa, pa, qa) {

var ra = [oa.author].concat(qa);

m.getMulti(ra, function(sa) {

if (oa.author == k.getID()) {

pa(r.\_("Du hast {subscriber1}, {subscriber2} und {subscriber3} hinzugef\u00fcgt.", [r.param("subscriber1", la(sa[qa[0]])), r.param("subscriber2", la(sa[qa[1]])), r.param("subscriber3", la(sa[qa[2]]))]));

} else if (qa[0] == k.getID()) {

pa(r.\_("{actor} hat dich, {subscriber2} und {subscriber3} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber2", la(sa[qa[1]])), r.param("subscriber3", la(sa[qa[2]]))]));

} else

pa(r.\_("{actor} hat {subscriber1}, {subscriber2} und {subscriber3} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber1", la(sa[qa[0]])), r.param("subscriber2", la(sa[qa[1]])), r.param("subscriber3", la(sa[qa[2]]))]));

});

}

function aa(oa, pa, qa) {

var ra = [oa.author].concat(qa);

m.getMulti(ra, function(sa) {

function ta(va) {

var wa = n.createElement("div", null, va.map(function(xa) {

return n.createElement("div", null, xa.name);

}));

return (n.createElement(p, {

alignH: "center",

position: "above",

tooltip: wa

}, r.\_("{num} weitere Personen", [r.param("num", va.length)])));

}

var ua = qa.map(function(va) {

return sa[va];

});

if (oa.author == k.getID()) {

pa(r.\_("Du hast {subscriber1}, {subscriber2} und {more\_people} hinzugef\u00fcgt.", [r.param("subscriber1", la(ua[0])), r.param("subscriber2", la(ua[1])), r.param("more\_people", ta(ua.slice(2)))]));

} else if (qa[0] == k.getID()) {

pa(r.\_("{actor} hat dich, {subscriber2} und {more\_people} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber2", la(ua[1])), r.param("more\_people", ta(ua.slice(2)))]));

} else

pa(r.\_("{actor} hat {subscriber1}, {subscriber2} und {more\_people} hinzugef\u00fcgt.", [r.param("actor", la(sa[oa.author])), r.param("subscriber1", la(ua[0])), r.param("subscriber2", la(ua[1])), r.param("more\_people", ta(ua.slice(2)))]));

});

}

function ba(oa, pa) {

var qa = [oa.author], ra = oa.log\_message\_data.removed\_participants, sa;

if (ra.length === 1) {

sa = ra[0];

qa.push(sa);

}

m.getMulti(qa, function(ta) {

var ua = ta[oa.author], va = ta[sa];

if (oa.author == k.getID()) {

if (!sa || sa == oa.author) {

pa("Du hast die Unterhaltung verlassen.");

} else

pa(r.\_("Du hast {name} von der Unterhaltung entfernt.", [r.param("name", la(va))]));

} else if (!sa || sa == oa.author) {

pa(r.\_("{actor} hat die Unterhaltung verlassen.", [r.param("actor", la(ua))]));

} else if (sa == k.getID()) {

pa(r.\_("{actor} hat dich von der Unterhaltung entfernt.", [r.param("actor", la(ua))]));

} else

pa(r.\_("{actor} hat {name} von der Unterhaltung entfernt.", [r.param("actor", la(ua)), r.param("name", la(va))]));

});

}

function ca(oa, pa) {

if (na(oa)) {

da(oa, pa);

return;

}

var qa = oa.log\_message\_data.caller, ra = oa.log\_message\_data.callee, sa = [qa, ra];

m.getMulti(sa, function(ta) {

var ua = ka(ta[ra]);

if (qa == k.getID()) {

if (oa.log\_message\_data.answered) {

pa(r.\_("Du hast {firstname} angerufen.", [r.param("firstname", ua)]));

} else

pa(r.\_("{firstname} hat einen Anruf von dir verpasst.", [r.param("firstname", ua)]));

} else {

var va = ka(ta[qa]);

if (oa.log\_message\_data.answered) {

pa(r.\_("{firstname} hat dich angerufen.", [r.param("firstname", va)]));

} else

pa(r.\_("Du hast einen Anruf von {firstname} verpasst.", [r.param("firstname", va)]));

}

});

}

function da(oa, pa) {

m.get(oa.log\_message\_data.callee, function(qa) {

var ra = ka(qa);

switch (oa.log\_message\_data.event\_name) {

case 'installing':

pa(r.\_("{firstname} installiert gerade Videoanrufe ...", [r.param("firstname", ra)]));

break;

case 'installed':

pa(r.\_("{firstname} hat Videoanrufe fertig installiert.", [r.param("firstname", ra)]));

break;

case 'install\_canceled':

pa("Du hast die Installation von Videoanrufen abgebrochen.");

break;

}

});

}

function ea(oa, pa) {

var qa = oa.log\_message\_data.caller, ra = oa.log\_message\_data.callee, sa = [qa, ra];

m.getMulti(sa, function(ta) {

if (qa == k.getID()) {

var ua = ka(ta[ra]);

if (oa.log\_message\_data.answered) {

pa(r.\_("Du hast {firstname} angerufen.", [r.param("firstname", ua)]));

} else

pa(r.\_("{firstname} hat einen Anruf von dir verpasst.", [r.param("firstname", ua)]));

} else {

var va = ka(ta[qa]);

if (oa.log\_message\_data.answered) {

pa(r.\_("{firstname} hat dich angerufen.", [r.param("firstname", va)]));

} else

pa(r.\_("Du hast einen Anruf von {firstname} verpasst.", [r.param("firstname", va)]));

}

});

}

function fa(oa, pa) {

var qa = oa.log\_message\_data.name;

if (oa.author == k.getID()) {

if (qa) {

pa(r.\_("Du hast die Unterhaltung benannt: {name}.", [r.param("name", n.createElement("span", {

className: "\_5wzs"

}, qa))]));

} else

pa("Du hast den Namen der Unterhaltung entfernt.");

} else

m.get(oa.author, function(ra) {

var sa = la(ra);

if (qa) {

pa(r.\_("{actor} hat der Unterhaltung einen Namen gegeben: {name}.", [r.param("actor", sa), r.param("name", n.createElement("span", {

className: "\_5wzs"

}, qa))]));

} else

pa(r.\_("{actor} hat den Namen der Unterhaltung entfernt.", [r.param("actor", sa)]));

});

}

function ga(oa, pa) {

if (oa.author == k.getID()) {

if (oa.log\_message\_data.image) {

pa("Du hast das Bild dieser Unterhaltung ge\u00e4ndert.");

} else

pa("Du hast das Bild der Unterhaltung entfernt.");

} else

m.get(oa.author, function(qa) {

var ra = la(qa);

if (oa.log\_message\_data.image) {

pa(r.\_("{actor} hat das Bild der Unterhaltung ge\u00e4ndert.", [r.param("actor", ra)]));

} else

pa(r.\_("{actor} hat das Bild der Unterhaltung entfernt.", [r.param("actor", ra)]));

});

}

function ha(oa, pa) {

if (oa.author == k.getID()) {

pa("Du hast den Hintergrund ge\u00e4ndert.");

} else

m.get(oa.author, function(qa) {

var ra = la(qa);

pa(r.\_("{actor} hat den Hintergrund ge\u00e4ndert.", [r.param("actor", ra)]));

});

}

function ia(oa, pa) {

pa("Couldn't find previous messages.");

}

function ja(oa, pa) {

var qa = oa.log\_message\_data, ra = qa.senderId, sa = qa.receiverId, ta = ra === k.getID();

m.getMulti([ra, sa], function(ua) {

pa(n.createElement(t, Object.assign({}, qa, {

receiverParticipant: ua[sa],

senderParticipant: ua[ra],

viewerIsSender: ta

})));

});

}

function ka(oa) {

return la(oa, true);

}

function la(oa, pa) {

var qa = oa.fbid, ra = j.constructEndpoint({

id: qa

});

if (oa.href)

return (n.createElement("a", {

className: "\_5wzt",

href: oa.href,

'data-hovercard': ra

}, pa ? oa.short\_name : oa.name));

return oa.name;

}

function ma(oa) {

var pa = oa.indexOf(k.getID());

if (pa > 0) {

var qa = oa.filter(function(ra) {

return ra !== k.getID();

});

return [k.getID()].concat(qa);

}

return oa;

}

function na(oa) {

return oa.log\_message\_data.event\_name === 'installing' || oa.log\_message\_data.event\_name === 'installed' || oa.log\_message\_data.event\_name === 'install\_canceled';

}

}, null);

\_\_d("MercuryLogMessage.react", ["DOM", "Event", "ChatVideoCallLink.react", "ImmutableObject", "MercuryLogMessageRenderer", "MercuryLogMessageType", "React", "ReactPropTypes", "VideoCallSupport", "cx", "emptyFunction", "formatDate", "joinClasses"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s) {

'use strict';

var t = m.createClass({

displayName: 'MercuryLogMessage',

propTypes: {

message: n.instanceOf(j).isRequired,

onImageLoad: n.func,

onVideoCallLinkClick: n.func

},

getDefaultProps: function() {

return {

onImageLoad: q

};

},

getInitialState: function() {

return {

messageText: null

};

},

componentDidMount: function() {

this.componentWillReceiveProps(this.props);

},

componentWillReceiveProps: function(u) {

k.renderText(u.message, function(v) {

this.setState({

messageText: v

});

}.bind(this));

},

shouldComponentUpdate: function(u, v) {

return this.state.messageText != v.messageText;

},

componentDidUpdate: function() {

var u = this.refs.attachment.getDOMNode();

k.renderAttachmentLegacy(u, this.props.message);

g.scry(u, 'img').forEach(function(v) {

var w = h.listen(v, 'load', function() {

this.props.onImageLoad(v);

w.remove();

}.bind(this));

}.bind(this));

},

render: function() {

return (m.createElement("div", Object.assign({}, this.props, {

className: s(this.props.className, "\_5ye6"),

title: r(new Date(this.props.message.timestamp), 'g:ia')

}), k.renderIcon(this.props.message), m.createElement("div", {

className: "\_5ye7"

}, this.state.messageText, this.renderVideoCallLink()), m.createElement("div", {

className: "\_5ye8",

ref: "attachment"

})));

},

renderVideoCallLink: function() {

if (this.props.message.log\_message\_type !== l.VIDEO\_CALL ||!this.props.onVideoCallLinkClick ||!o.isVideoCallSupported())

return null;

return (m.createElement(i, {

message: this.props.message,

onClick: this.props.onVideoCallLinkClick

}));

}

});

e.exports = t;

}, null);

\_\_d("ChatConversation.react", ["ChatDateBreak.react", "ChatMessage.react", "ImmutableObject", "MercuryActionTypeConstants", "MercuryLogMessage.react", "React", "cx", "invariant", "OrionNUXRenderer"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n) {

'use strict';

var o = b('OrionNUXRenderer').module, p = 1000 \* 60 \* 60, q = l.createClass({

displayName: 'ChatConversation',

propTypes: {

maxBubbleWidth: l.PropTypes.number.isRequired,

messages: l.PropTypes.arrayOf(l.PropTypes.instanceOf(i)).isRequired,

onCallLinkClick: l.PropTypes.func,

onImageLoad: l.PropTypes.func,

onStickerClick: l.PropTypes.func

},

componentWillReceiveProps: function(r) {

n(r.maxBubbleWidth === this.props.maxBubbleWidth);

},

render: function() {

var r = this.props.messages, s = [];

for (var t = 0; t < r.length; t++) {

var u = r[t], v = (t > 0) ? r[t - 1]: null;

if (u.is\_cleared)

continue;

if (!v || v.is\_cleared || u.timestamp - v.timestamp > p)

s.push(l.createElement(g, {

date: new Date(u.timestamp),

key: 'db:' + u.message\_id

}));

if (u.action\_type == j.LOG\_MESSAGE) {

s.push(l.createElement(k, {

className: "\_5w0o",

key: u.message\_id,

message: u,

onImageLoad: this.props.onImageLoad,

onCallLinkClick: this.props.onCallLinkClick

}));

} else {

s.push(l.createElement(h, {

key: u.message\_id,

message: u,

maxBubbleWidth: this.props.maxBubbleWidth,

onImageLoad: this.props.onImageLoad,

onStickerClick: this.props.onStickerClick

}));

if (o && o.isValidNUXMessage(u))

s.push(l.createElement(o, {

key: 'onr\_c:' + u.message\_id,

message: u,

onImageLoad: this.props.onImageLoad,

type: "chat"

}));

}

}

return l.createElement("div", null, s);

}

});

e.exports = q;

}, null);

\_\_d("MercuryTypingReceiver", ["Arbiter", "ChannelConstants", "MercuryActionTypeConstants", "MercuryIDs", "MercuryPayloadSource", "MercuryViewer", "TypingStates", "mixInEventEmitter", "setTimeoutAcrossTransitions", "MercuryServerRequests"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o) {

var p = b('MercuryServerRequests').get(), q, r = {}, s = 30000, t = {};

n(t, {

'state-changed': true

});

function u(aa) {

var ba = r[aa] || {}, ca = Object.keys(ba);

ca.sort(function(da, ea) {

return ba[da] - ba[ea];

});

return ca;

}

function v() {

q = null;

var aa = Date.now(), ba = {}, ca = false;

for (var da in r) {

var ea = false;

for (var fa in r[da] || {})

if (r[da][fa] < aa - s) {

delete r[da][fa];

ea = true;

} else

ca = true;

if (ea)

ba[da] = u(da);

}

for (var ga in ba) {

x(ba);

break;

}

if (ca)

q = o(v, 3000);

}

function w(aa, ba) {

if (aa in r)

if (ba in r[aa]) {

delete r[aa][ba];

y(aa);

}

}

function x(aa) {

t.releaseHeldEventType('state-changed');

t.emitAndHold('state-changed', aa);

}

function y(aa) {

var ba = {};

ba[aa] = u(aa);

x(ba);

}

function z(aa) {

if (aa.thread\_fbid)

return p.getClientThreadIDNow(aa.thread\_fbid);

if (aa.type === 'typ')

return j.getThreadIDFromUserID(aa.from);

return null;

}

g.subscribe([h.getArbiterType('typ'), h.getArbiterType('ttyp')], function(aa, ba) {

var ca = ba.obj, da = z(ca);

if (da) {

var ea = j.getParticipantIDFromUserID(ca.from);

if (ca.st == m.TYPING) {

r[da] = r[da] || {};

var fa = r[da][ea];

r[da][ea] = Date.now();

if (!q)

q = o(v, 3000);

!fa && y(da);

} else if (ca.st == m.INACTIVE)

w(da, ea);

}

});

p.subscribe('update-typing-state', function(aa, ba) {

var ca = ba.payload\_source;

if (ca != k.CLIENT\_CHANNEL\_MESSAGE)

return;

var da = ba.actions;

if (!da ||!da.length)

return;

var ea = i.USER\_GENERATED\_MESSAGE;

da.forEach(function(fa) {

if (fa.action\_type == ea && fa.author != l.getID())

w(fa.thread\_id, fa.author);

});

});

e.exports = t;

}, null);

\_\_d("ChatTypingIndicators.react", ["ChatAuthorPhotoBlock.react", "DOM", "MercuryIDs", "MercuryParticipants", "ReactPropTypes", "React", "SubscriptionsHandler", "Tooltip", "MercuryTypingReceiver", "arraySort", "createObjectFrom", "cx", "emptyFunction", "fbt", "MercuryThreadInformer"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t) {

var u = b('MercuryThreadInformer').get(), v = l.createClass({

displayName: 'ChatTypingIndicators',

propTypes: {

indicatorsWillShow: k.func,

indicatorsDidShow: k.func,

threadID: k.string.isRequired

},

getDefaultProps: function() {

return {

indicatorsWillShow: s,

indicatorsDidShow: s

};

},

getInitialState: function() {

return {

typingUserIDs: []

};

},

componentDidMount: function() {

this.\_subscriptions = new m();

this.\_subscriptions.addSubscriptions(o.addRetroactiveListener('state-changed', this.typingStateChanged), u.subscribe('messages-received', this.messagesReceived));

},

componentWillReceiveProps: function(w) {

if (w.threadID != this.props.threadID)

this.setState({

typingUserIDs: []

});

},

componentWillUpdate: function(w, x) {

if (x.typingUserIDs.length > 0)

this.props.indicatorsWillShow();

},

componentDidUpdate: function() {

if (this.state.typingUserIDs.length > 0)

this.props.indicatorsDidShow();

j.getMulti(this.state.typingUserIDs, function(w) {

if (this.isMounted())

this.state.typingUserIDs.forEach(function(x) {

var y = w[x];

n.set(this.refs[x].getDOMNode(), this.renderTooltip(y.short\_name), 'above', 'left');

}.bind(this));

}.bind(this));

},

componentWillUnmount: function() {

this.\_subscriptions.release();

},

render: function() {

var w = i.isMultichat(this.props.threadID);

return (l.createElement("div", {

className: "\_2fsr"

}, this.state.typingUserIDs.map(function(x) {

return l.createElement(g, {

authorID: x,

className: (("\_gfq") + (w ? ' ' + "\_52fu" : '')),

hideName: !w,

key: x

}, l.createElement("div", {

className: "\_52ft"

}, l.createElement("div", {

className: "\_gfp",

ref: x

})));

})));

},

renderTooltip: function(w) {

var x = h.create('span');

l.render(l.createElement("span", null, t.\_("{name} schreibt etwas ...", [t.param("name", w)])), x);

return x;

},

typingStateChanged: function(w) {

if (this.props.threadID in w)

this.setState({

typingUserIDs: p(w[this.props.threadID])

});

},

messagesReceived: function(w, x) {

if (this.props.threadID in x) {

var y = x[this.props.threadID], z = q(y.map(function(aa) {

return aa.author;

}));

this.setState({

typingUserIDs: p(this.state.typingUserIDs.filter(function(aa) {

return !z[aa];

}))

});

}

}

});

e.exports = v;

}, null);

\_\_d("MercuryIndicatorController", ["ArbiterMixin", "DOM", "MercuryActionTypeConstants", "MercuryDelayedRoger", "MercuryIDs", "MercuryMessageSourceTags", "MercuryParticipants", "MercuryRoger", "MercuryTypingReceiver", "MercuryViewer", "arrayContains", "copyProperties", "formatDate", "removeFromArray", "tx", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u) {

var v = b('MercuryThreads').get(), w = [];

function x(y) {

this.\_threadID = y;

this.\_canonicalUser = k.getUserIDFromThreadID(y);

w.push(this);

}

r(x.prototype, g, {

destroy: function() {

t(w, this);

},

setLastMessage: function(y) {

this.\_lastMsg = y;

this.\_handleStateChange();

},

\_informStateChanged: function(y) {

if (y.activity == 'none' && this.\_currentActivity == 'none')

return;

if (this.\_lastMsg && p.isViewer(this.\_lastMsg.author))

y.self\_authored = true;

this.\_currentActivity = y.activity;

this.inform('state-changed', y);

},

\_notifySentFrom: function() {

var y, z, aa = this.\_lastMsg.location\_text, ba = this.\_lastMsg.source\_tags || [];

if (aa) {

y = u.\_("Gesendet aus {location}", {

location: aa

});

z = 'sentFromMobile';

} else if (q(ba, l.MESSENGER)) {

y = h.create('a', {

href: '/mobile/messenger',

'class': 'fcg',

target: '\_blank'

}, "Vom Messenger gesendet");

z = 'sentFromMobile';

} else if (q(ba, l.MOBILE)) {

y = h.create('a', {

href: '/mobile',

'class': 'fcg',

target: '\_blank'

}, "Vom Handy gesendet");

z = 'sentFromMobile';

} else if (q(ba, l.EMAIL)) {

y = "Per E-Mail gesendet";

z = 'sentFromEmail';

} else {

this.\_informStateChanged({

activity: 'none'

});

return;

}

this.\_informStateChanged({

activity: z,

text: y

});

},

\_notifySeenTimestamp: function(y) {

var z = n.getSeenTimestamp(this.\_threadID, y[0]) \* .001, aa = Date.now() \* .001, ba;

if (z < aa - 518400) {

ba = 'M j';

} else if (z < aa - 86400) {

ba = 'D g:ia';

} else

ba = 'g:ia';

this.\_informStateChanged({

activity: 'seen-timestamp',

text: u.\_("Gesehen: {timestamp}", {

timestamp: s(z, ba)

})

});

},

\_nameNormalizer: function(y) {

var z;

m.getMulti(y, function(aa) {

function ba(da) {

if (aa[da] !== undefined) {

return aa[da].short\_name.toLowerCase();

} else

return da;

}

var ca = y.map(ba);

z = function(da) {

var ea = ba(da), fa = ca.indexOf(ea) !== ca.lastIndexOf(ea);

return fa ? aa[da].name : aa[da].short\_name;

};

});

return z;

},

\_notifySeenBy: function(y) {

var z = this.\_lastMsg, aa = true;

m.getMulti(y, function(ba) {

aa = false;

if (this.\_lastMsg != z)

return;

var ca = v.getThreadMetaNow(this.\_threadID), da = ca ? ca.participants.length: 0, ea = y.length + (z.author != p.getID()), fa, ga = false, ha = da > 2 && ea >= da - 1, ia;

if (ca)

ia = this.\_nameNormalizer(ca.participants);

if (!ia)

ia = function(ma) {

return ba[ma].short\_name;

};

if (ha) {

fa = "Von allen gesehen";

} else if (y.length == 1) {

fa = u.\_("Gesehen von {user}", {

user: ia(y[0])

});

} else if (y.length == 2) {

fa = u.\_("Gesehen von {user1}, {user2}", {

user1: ia(y[0]),

user2: ia(y[1])

});

} else if (y.length == 3) {

fa = u.\_("Gesehen von {user1}, {user2}, {user3}", {

user1: ia(y[0]),

user2: ia(y[1]),

user3: ia(y[2])

});

} else if (y.length > 3) {

var ja = Object.keys(ba).length - 2, ka = u.\_("{num} weiteren Personen", {

num: ja

}), la = h.create('span', {

className: 'more'

}, ka);

fa = h.tx.\_("Gesehen von {user1}, {user2}, {=num more link}", {

user1: ia(y[0]),

user2: ia(y[1]),

'=num more link': la

});

ga = true;

}

this.\_informStateChanged({

activity: 'seen-by',

text: fa,

seenBy: y,

tooltip: ga

});

}.bind(this));

aa && this.\_informStateChanged({

activity: 'none'

});

},

\_notifyTyping: function(y) {

var z = this.\_lastMsg, aa = true;

m.getMulti(y, function(ba) {

aa = false;

if (this.\_lastMsg != z)

return;

var ca = v.getThreadMetaNow(this.\_threadID), da = ca ? ca.participants.length: 0, ea;

if (ca)

ea = this.\_nameNormalizer(ca.participants);

if (!ea)

ea = function(ka) {

return ba[ka].short\_name;

};

var fa, ga = false;

if (da > 2 && y.length >= da - 1) {

fa = "Alle schreiben etwas ...";

} else if (y.length == 1) {

fa = u.\_("{name} schreibt etwas ...", {

name: ea(y[0])

});

} else if (y.length == 2) {

fa = u.\_("{user1} und {user2} schreiben etwas ...", {

user1: ea(y[0]),

user2: ea(y[1])

});

} else if (y.length == 3) {

fa = u.\_("{user1}, {user2} und {user3} schreiben etwas ...", {

user1: ea(y[0]),

user2: ea(y[1]),

user3: ea(y[2])

});

} else if (y.length > 3) {

var ha = Object.keys(ba).length - 2, ia = u.\_("{num} weitere Personen", {

num: ha

}), ja = h.create('a', {

href: '#'

}, ia);

fa = h.tx.\_("{user1}, {user2} und {=num more link} schreiben etwas ...", {

user1: ea(y[0]),

user2: ea(y[1]),

'=num more link': ja

});

ga = true;

}

this.\_informStateChanged({

activity: 'typing',

text: fa,

typing: y,

tooltip: ga

});

}.bind(this));

aa && this.\_informStateChanged({

activity: 'none'

});

},

\_handleStateChange: function() {

var y = i.LOG\_MESSAGE;

if (!this.\_lastMsg || this.\_lastMsg.action\_type == y) {

this.\_informStateChanged({

activity: 'none'

});

return;

}

if (this.\_typing && this.\_typing.length) {

this.\_notifyTyping(this.\_typing);

return;

}

if (this.\_canonicalUser && this.\_lastMsg.author != p.getID()) {

this.\_notifySentFrom();

return;

}

var z = j.getSeenBy(this.\_threadID, true);

if (z.length)

if (this.\_canonicalUser) {

this.\_notifySeenTimestamp(z);

return;

} else {

this.\_notifySeenBy(z);

return;

}

this.\_informStateChanged({

activity: 'none'

});

}

});

o.addRetroactiveListener('state-changed', function(y) {

w.forEach(function(z) {

var aa = y[z.\_threadID];

if (aa !== undefined) {

z.\_typing = aa;

z.\_handleStateChange();

}

});

});

j.subscribe('state-changed', function(y, z) {

w.forEach(function(aa) {

z[aa.\_threadID] && aa.\_handleStateChange();

});

});

e.exports = x;

}, null);

\_\_d("MercuryLastMessageIndicator.react", ["DOM", "MercuryIndicatorController", "MercuryParticipants", "ReactPropTypes", "React", "Tooltip", "cx", "emptyFunction", "joinClasses", "startsWith"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

var q = k.createClass({

displayName: 'MercuryLastMessageIndicator',

propTypes: {

hideTyping: j.bool,

indicatorWillShow: j.func,

indicatorDidShow: j.func,

lastMessage: j.object,

threadID: j.string.isRequired

},

getDefaultProps: function() {

return {

indicatorWillShow: n,

indicatorDidShow: n

};

},

getInitialState: function() {

return {

data: {}

};

},

componentDidMount: function() {

this.\_setup(this.props);

},

componentWillReceiveProps: function(r) {

if (r.threadID != this.props.threadID) {

this.\_destroy();

this.\_setup(r);

} else if (r.lastMessage != this.props.lastMessage)

this.\_controller.setLastMessage(r.lastMessage);

},

componentWillUpdate: function(r, s) {

if (this.isVisible(r, s))

this.props.indicatorWillShow();

},

componentDidUpdate: function() {

if (!this.isVisible())

return;

this.setText();

this.setTooltip();

this.props.indicatorDidShow();

},

componentWillUnmount: function() {

this.\_destroy();

},

render: function() {

return (k.createElement("div", {

className: this.getRootClass()

}, k.createElement("div", {

className: "\_510h"

}), k.createElement("span", {

className: "\_510f",

ref: "text"

}, " ")));

},

getRootClass: function() {

var r = (("\_510g") + (this.state.data.self\_authored ? ' ' + "\_510e" : '')), s = this.state.data.activity, t = null;

if (p(s, 'seen')) {

t = 'seen';

} else if (s == 'typing') {

t = this.props.hideTyping ? null : s;

} else

t = s;

return o(r, t, this.props.className);

},

isVisible: function(r, s) {

r = r || this.props;

s = s || this.state;

return (s.data && s.data.activity != 'none'&&!(r.hideTyping && s.data.activity == 'typing'));

},

setText: function() {

if (this.state.data.text)

g.setContent(this.refs.text.getDOMNode(), this.state.data.text);

},

setTooltip: function() {

if (this.state.data.activity == 'seen-by' && this.state.data.tooltip)

i.getMulti(this.state.data.seenBy, function(r) {

var s = g.create('div', null, this.state.data.seenBy.map(function(t) {

return g.create('div', null, r[t].name);

}));

l.set(g.find(this.getDOMNode(), 'span.more'), s, 'above', 'center');

}.bind(this));

},

\_setup: function(r) {

this.\_controller = new h(r.threadID);

this.\_subscription = this.\_controller.subscribe('state-changed', function(s, t) {

return this.setState({

data: t

});

}.bind(this));

r.lastMessage && this.\_controller.setLastMessage(r.lastMessage);

},

\_destroy: function() {

this.\_subscription.unsubscribe();

this.\_controller.destroy();

},

setLastMessage: function(r) {

this.setProps({

lastMessage: r

});

}

});

e.exports = q;

}, null);

\_\_d("MercuryTypingIndicator", ["Animation", "ChatConfig", "CSS", "DOM", "MercuryTypingAnimation.react", "MercuryTypingReceiver", "MercuryViewer", "MercuryParticipants", "React", "Style", "ChatTabTemplates", "Tooltip", "copyProperties", "csx", "cx", "tx", "MercuryThreadInformer"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v) {

var w = b('MercuryThreadInformer').get(), x = [];

w.subscribe('messages-received', function(aa, ba) {

x.forEach(function(ca) {

var da = ba[ca.\_threadID];

da && ca.receivedMessages(da);

});

});

l.addRetroactiveListener('state-changed', function(aa) {

x.forEach(function(ba) {

var ca = aa[ba.\_threadID];

ca && ba.\_handleStateChanged(ca);

});

});

function y(aa) {

var ba = q[':fb:chat:conversation:message-group'].build(), ca = q[':fb:mercury:typing-indicator:typing'].build();

i.addClass(ba.getRoot(), "\_50kd");

var da = ba.getNode('profileLink');

r.set(da, aa.name, 'left');

da.href = aa.href;

ba.setNodeContent('profileName', aa.name);

ba.setNodeProperty('profilePhoto', 'src', aa.image\_src);

var ea = v.\_("{name} schreibt etwas ...", {

name: aa.short\_name

});

r.set(ca.getRoot(), ea, 'above');

j.appendContent(ba.getNode('messages'), ca.getRoot());

return ba;

}

function z(aa, ba, ca) {

this.\_animations = {};

this.\_activeUsers = {};

this.\_typingIndicator = ba;

this.\_messagesView = ca;

this.\_threadID = aa;

this.\_subscription = l.addRetroactiveListener('state-changed', function(da) {

var ea = da[this.\_threadID];

ea && this.\_handleStateChanged(ea);

}.bind(this));

x.push(this);

}

s(z.prototype, {

destroy: function() {

Object.keys(this.\_activeUsers).forEach(this.\_removeUserBubble.bind(this));

this.\_controller.destroy();

x.remove(this);

},

receivedMessages: function(aa) {

aa.forEach(function(ba) {

if (!m.isViewer(ba.author))

this.\_removeUserBubble(ba.author);

}.bind(this));

},

\_handleStateChanged: function(aa) {

for (var ba in this.\_activeUsers)

if (aa.indexOf(ba)===-1) {

this.\_slideOutUserBubble(ba);

delete this.\_activeUsers[ba];

}

if (aa.length)

n.getMulti(aa, function(ca) {

var da = this.\_messagesView.isScrolledToBottom(), ea = {};

for (var fa in ca) {

var ga = this.\_activeUsers[fa];

ea[fa] = ga || y(ca[fa]).getRoot();

if (!ga) {

j.appendContent(this.\_typingIndicator, ea[fa]);

if (h.get('chat\_thread\_typing\_indicator\_animated')) {

var ha = j.scry(this.\_typingIndicator, ".\_510u");

for (var ia = 0, ja = ha.length; ia < ja; ia++)

o.render(o.createElement(k, {

className: "\_3e2s"

}), ha[ia]);

}

}

}

var ka = Object.keys(ea).length > 0;

da && this.\_messagesView.scrollToBottom(ka);

this.\_activeUsers = ea;

}.bind(this));

},

\_removeUserBubble: function(aa, ba) {

var ca = this.\_getCurrentAnimation(aa, ba);

if (ca) {

ca.animation.stop();

j.remove(ca.elem);

delete this.\_animations[aa];

}

if (aa in this.\_activeUsers) {

j.remove(this.\_activeUsers[aa]);

delete this.\_activeUsers[aa];

}

if (ba && h.get('chat\_thread\_typing\_indicator\_animated')) {

var da = j.scry(ba, ".\_510u");

for (var ea = 0, fa = da.length; ea < fa; ea++)

o.unmountComponentAtNode(da[ea]);

}

ba && j.remove(ba);

},

\_slideOutUserBubble: function(aa) {

var ba = this.\_activeUsers[aa];

if (this.\_getCurrentAnimation(aa, ba)) {

return;

} else if (ba) {

p.set(ba, 'overflow', 'hidden');

var ca = (new g(ba)).from('opacity', 1).from('height', ba.offsetHeight).to('height', 0).to('opacity', 0).ease(g.ease.end).duration(250).ondone(this.\_removeUserBubble.bind(this, aa, ba)).go();

this.\_animations[aa] = {

animation: ca,

elem: ba

};

}

},

\_getCurrentAnimation: function(aa, ba) {

if (this.\_animations[aa] && (!ba || this.\_animations[aa].elem === ba))

return this.\_animations[aa];

}

});

e.exports = z;

}, null);

\_\_d("MercuryMessageList", ["ImmutableObject", "LogHistory", "OrderedMap", "setTimeoutAcrossTransitions"], function(a, b, c, d, e, f, g, h, i, j) {

'use strict';

var k = h.getInstance('mercury\_message\_list');

function l(q) {

this.$MercuryMessageList0 = n(q ? m(q) : []);

}

l.prototype.append = function(q) {

this.$MercuryMessageList0 = this.$MercuryMessageList0.merge(n(m(q)));

p('append', q);

};

l.prototype.prepend = function(q) {

this.$MercuryMessageList0 = n(m(q)).merge(this.$MercuryMessageList0);

p('prepend', q);

};

l.prototype.update = function(q) {

var r = q.filter(function(s) {

return this.contains(s);

}.bind(this));

if (r.length === 0)

return;

this.$MercuryMessageList0 = this.$MercuryMessageList0.merge(n(m(r)));

p('update', q);

};

l.prototype.reorder = function(q) {

this.$MercuryMessageList0 = n(q.map(function(r) {

var s = this.$MercuryMessageList0.get(r.message\_id);

if (!s || s.timestamp != r.timestamp || s.is\_cleared != r.is\_cleared)

return new g(r);

return s;

}.bind(this)));

p('reorder', q);

};

l.prototype.hasReachedClearedMessages = function() {

var q = this.messageAt(0);

return !!(q && q.is\_cleared);

};

l.prototype.messageAt = function(q) {

var r = this.$MercuryMessageList0.keyAtIndex(q);

return r ? this.$MercuryMessageList0.get(r) : undefined;

};

l.prototype.contains = function(q) {

return this.$MercuryMessageList0.indexOfKey(o(q)) !== undefined;

};

l.prototype.toArray = function() {

return this.$MercuryMessageList0.toArray();

};

l.prototype.toOrderedMap = function() {

return this.$MercuryMessageList0;

};

function m(q) {

return q.map(function(r) {

return new g(r);

});

}

function n(q) {

return i.fromArray(q, o);

}

function o(q) {

return q.message\_id;

}

function p(event, q) {

j(function() {

k.debug(event, {

messageIDs: q.map(function(r) {

return r.message\_id;

})

});

}, 0);

}

e.exports = l;

}, null);

\_\_d("MercuryMessageStore", ["MercuryAPIArgsSource", "LogHistory", "MercuryViewer", "MercuryMessageList", "SubscriptionsHandler", "MercuryThreadlistConstants", "copyProperties", "merge", "mixInEventEmitter", "setTimeoutAcrossTransitions", "MercuryMessages", "MercuryThreadInformer", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

'use strict';

var q = b('MercuryMessages').get(), r = b('MercuryThreadInformer').get(), s = b('MercuryThreads').get(), t = h.getInstance('mercury\_message\_store');

function u(w) {

this.$MercuryMessageStore0 = w;

this.$MercuryMessageStore1 = new j();

this.$MercuryMessageStore2 = 1;

this.$MercuryMessageStore3 = null;

this.$MercuryMessageStore4 = 0;

this.$MercuryMessageStore5 = new k();

this.$MercuryMessageStore5.addSubscriptions(r.subscribe('messages-received', function(x, y) {

return this.$MercuryMessageStore7(y);

}.bind(this)), r.subscribe('messages-updated', function(x, y) {

return y[this.$MercuryMessageStore0] && this.$MercuryMessageStore8(y[this.$MercuryMessageStore0]);

}.bind(this)), r.subscribe('messages-reordered', function(x, y) {

return y[this.$MercuryMessageStore0] && this.$MercuryMessageStore9();

}.bind(this)));

t.debug('constructed', {

threadID: this.$MercuryMessageStore0

});

this.$MercuryMessageStorea();

}

u.prototype.setNewThreadID = function(w) {

this.$MercuryMessageStoreb = w;

};

u.prototype.destroy = function() {

this.$MercuryMessageStore5 && this.$MercuryMessageStore5.release();

t.debug('destroyed', {

threadID: this.$MercuryMessageStore0

});

};

u.prototype.subscribe = function(w) {

return this.addRetroactiveListener('updated', w);

};

u.prototype.fetchMoreMessages = function() {

if (q.hasLoadedExactlyNMessages(this.$MercuryMessageStore0, this.$MercuryMessageStore4) && q.hasLoadedAllMessages(this.$MercuryMessageStore0))

return false;

if (s.isNewEmptyLocalThread(this.$MercuryMessageStore0))

return false;

if (this.$MercuryMessageStore1.hasReachedClearedMessages())

return false;

return this.$MercuryMessageStorea();

};

u.prototype.$MercuryMessageStorea = function() {

var w = this.$MercuryMessageStore4 + l.RECENT\_MESSAGES\_LIMIT \* this.$MercuryMessageStore2;

t.debug('fetching', {

threadID: this.$MercuryMessageStore0,

limit: w

});

q.getThreadMessagesRange(this.$MercuryMessageStore0, 0, w, this.$MercuryMessageStorec.bind(this), null, g.MERCURY);

if (this.$MercuryMessageStore2 < 10)

this.$MercuryMessageStore2 += 1;

if (q.hasLoadedNMessages(this.$MercuryMessageStore0, w)) {

this.$MercuryMessageStore9();

return false;

}

return true;

};

u.prototype.$MercuryMessageStorec = function(w) {

if (w && w.length) {

this.$MercuryMessageStore3 = w[0].timestamp;

this.$MercuryMessageStore4 = w.length;

t.debug('fetched', {

threadID: this.$MercuryMessageStore0,

earliestMessageTimestamp: this.$MercuryMessageStore3,

count: this.$MercuryMessageStore4

});

}

};

u.prototype.$MercuryMessageStore7 = function(w) {

var x = [], y = w[this.$MercuryMessageStore0];

if (y && y.length)

x = x.concat(y);

if (this.$MercuryMessageStoreb) {

var z = w[this.$MercuryMessageStoreb];

if (z && z.length) {

x = x.concat(z);

this.$MercuryMessageStoreb = null;

}

}

if (x.length) {

this.$MercuryMessageStore1.append(x);

this.$MercuryMessageStore4 += x.length;

var aa = x.every(function(ba) {

return ba.author != i.getID();

});

this.$MercuryMessageStored(u.MESSAGES\_RECEIVED, {

allFromOthers: aa

});

}

};

u.prototype.$MercuryMessageStore8 = function(w) {

this.$MercuryMessageStore1.update(q.getMessagesFromIDs(Object.keys(w)));

this.$MercuryMessageStoree();

this.$MercuryMessageStored(u.MESSAGES\_CHANGED);

};

u.prototype.$MercuryMessageStore9 = function() {

if (!this.$MercuryMessageStore3)

return;

this.$MercuryMessageStore1.reorder(q.getThreadMessagesSinceTimestamp(this.$MercuryMessageStore0, this.$MercuryMessageStore3));

this.$MercuryMessageStoree();

this.$MercuryMessageStored(u.MESSAGES\_REORDERED);

};

u.prototype.$MercuryMessageStored = function(w, x) {

var y = this.$MercuryMessageStore1.toArray();

this.releaseHeldEventType('updated');

this.emitAndHold('updated', n({

messages: y,

eventType: w

}, x));

v(this.$MercuryMessageStore0, w, y);

};

u.prototype.$MercuryMessageStoree = function() {

var w = this.$MercuryMessageStore1.messageAt(0);

if (w)

this.$MercuryMessageStore3 = w.timestamp;

};

function v(w, x, y) {

p(function() {

t.debug('update:' + x, {

threadID: w,

messageIDs: y.map(function(z) {

return z.message\_id;

})

});

}, 0);

}

m(u, {

MESSAGES\_FETCHED: 'fetched',

MESSAGES\_CHANGED: 'changed',

MESSAGES\_RECEIVED: 'received',

MESSAGES\_REORDERED: 'reordered'

});

o(u, {

updated: true

});

e.exports = u;

}, null);

\_\_d("ChatTabMessagesView", ["Animation", "Arbiter", "ArbiterMixin", "BanzaiLogger", "ChatConfig", "ChatConversation.react", "ChatTypingIndicators.react", "ChatWelcomeMessage", "CSS", "DOM", "Event", "MercuryLastMessageIndicator.react", "LiveTimer", "MercuryTypingIndicator", "MercuryViewer", "MercuryMessageStore", "React", "ServerTime", "StickerActions", "StickerState", "StickerStoreController", "Style", "SubscriptionsHandler", "UserAgent\_DEPRECATED", "arrayContains", "copyProperties", "getElementPosition", "throttle"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, aa, ba, ca, da, ea, fa, ga, ha) {

'use strict';

var ia = 70, ja = null, ka = 20;

function la(na) {

if (ja === null) {

var oa = na.childNodes[0];

ba.set(na, 'overflow', 'scroll');

ja = oa.clientWidth - ia;

ba.set(na, 'overflow', '');

}

return ja;

}

function ma(na, oa, pa, qa, ra, sa, ta, ua) {

this.loadingIcon = ra;

this.threadID = na;

this.sheetController = oa;

this.scrollContainer = pa;

this.conversationElem = qa;

this.tabView = ua;

s.restart(x.get() / 1000);

this.\_loadingMoreMessages = false;

function va() {

ma.inform('interaction-with-tab', na);

}

this.\_subscriptions = new ca();

this.\_subscriptions.addSubscriptions(h.subscribe('overflow-applied-to-body', this.scrollToBottom.bind(this)), q.listen(this.scrollContainer, 'mousedown', va));

if (da.firefox()) {

var wa = ('WheelEvent' in window) ? 'wheel': 'DOMMouseScroll';

this.scrollContainer.addEventListener(wa, va, false);

} else

this.\_subscriptions.addSubscriptions(q.listen(this.scrollContainer, 'mousewheel', va));

this.\_subscriptions.addSubscriptions(q.listen(this.scrollContainer, 'scroll', ha(this.scrolling, 50, this)));

if (k.get('chat\_react')) {

var xa;

w.render(w.createElement(m, {

threadID: this.threadID,

indicatorsWillShow: function() {

return xa = this.isScrolledToBottom();

}.bind(this),

indicatorsDidShow: function() {

return xa && this.scrollToBottom(true);

}.bind(this)

}), ta);

} else

this.typingIndicator = new t(this.threadID, ta, this);

var ya;

this.lastMessageIndicatorNode = sa;

this.lastMessageIndicator = w.render(w.createElement(r, {

threadID: this.threadID,

hideTyping: true,

indicatorWillShow: function() {

return ya = this.isScrolledToBottom();

}.bind(this),

indicatorDidShow: function() {

return ya && this.scrollToBottom(true);

}.bind(this)

}), this.lastMessageIndicatorNode);

this.initializeConversation();

}

fa(ma, i);

fa(ma.prototype, {

initializeConversation: function() {

this.\_store = new v(this.threadID);

this.\_conversation = w.render(w.createElement(l, {

maxBubbleWidth: la(this.scrollContainer),

messages: [],

onImageLoad: function(na) {

var oa = this.scrollContainer, pa = oa.scrollTop + oa.clientHeight;

if (pa + na.offsetHeight >= oa.scrollHeight)

this.scrollToBottom();

}.bind(this),

onStickerClick: this.\_onStickerClick.bind(this),

onCallLinkClick: function(na, oa, pa) {

ma.inform('video-call-clicked', {

userID: na,

threadID: oa,

clickSource: pa

});

}

}), this.conversationElem);

this.\_subscriptions.addSubscriptions(this.\_store.subscribe(function(na) {

if (na.eventType == v.MESSAGES\_RECEIVED && na.allFromOthers&&!this.isScrolledToBottom()) {

this.sheetController.openNewMessagesSheet();

this.\_newMessagesSheetOpened = true;

}

if (na.eventType == v.MESSAGES\_REORDERED) {

this.\_loadingMoreMessages = false;

o.hide(this.loadingIcon);

}

var oa = this.isScrolledToBottom(), pa = this.\_getLoadingHeight(), qa = this.scrollContainer.scrollHeight, ra = this.scrollContainer.scrollTop, sa = na.messages, ta = n.getWelcomeMessage(this.threadID);

if (ta)

sa.push(ta);

this.\_conversation.setProps({

messages: sa

}, function() {

if (oa || this.\_shouldScrollToBottom) {

this.scrollToBottom();

this.setShouldScrollToBottom(false);

} else if (na.eventType == v.MESSAGES\_REORDERED)

this.scrollToPosition(this.scrollContainer.scrollHeight - qa - pa + ra);

}.bind(this));

this.lastMessageIndicator.setProps({

lastMessage: sa.length > 0 ? sa[sa.length - 1]: null

});

if (na.eventType == v.MESSAGES\_RECEIVED && this.\_shouldShowStickerReplyNUX(sa) && this.tabView && this.tabView.isFocused())

this.tabView.showStickerReplyNUX();

}.bind(this)));

},

setShouldScrollToBottom: function(na) {

this.\_shouldScrollToBottom = na;

},

scrolling: function() {

this.\_checkToAnimateSticker();

if (this.isScrolledNearTop()&&!this.\_loadingMoreMessages&&!this.isScrolledToBottom()&&!this.tabView.\_isDragDropActive)

if (this.\_store.fetchMoreMessages()) {

o.show(this.loadingIcon);

this.\_loadingMoreMessages = true;

}

if (!this.\_newMessagesSheetOpened)

return;

if (this.isScrolledToBottom()) {

this.sheetController.closeNewMessagesSheet();

this.\_newMessagesSheetOpened = false;

}

},

getScrollTop: function() {

return this.scrollContainer && this.scrollContainer.scrollTop;

},

destroy: function() {

p.scry(this.conversationElem, '.stickerContainer').forEach(function(na) {

return w.unmountComponentAtNode(na);

});

w.unmountComponentAtNode(this.conversationElem);

this.\_subscriptions && this.\_subscriptions.release();

w.unmountComponentAtNode(this.lastMessageIndicatorNode);

delete this.lastMessageIndicator;

this.\_store && this.\_store.destroy();

this.destroyed = true;

},

\_getLoadingHeight: function() {

return this.loadingHeight || this.loadingIcon.clientHeight;

},

\_shouldShowStickerReplyNUX: function(na) {

if (!z.shouldShowStickerReplyNUX() ||!na.length)

return false;

var oa = na[na.length - 1];

if (oa.author === u.getID())

return false;

if (oa.has\_attachment)

return oa.attachments.some(function(pa) {

return pa.attach\_type === 'sticker';

});

return false;

},

isScrolledToBottom: function(na) {

var oa = this.scrollContainer;

na = na || ka;

return oa.scrollTop + oa.clientHeight >= oa.scrollHeight - na;

},

isScrolledNearTop: function() {

return this.scrollContainer.scrollTop < this.scrollContainer.clientHeight;

},

scrollToBottom: function(na) {

this.scrollToPosition(this.scrollContainer.scrollHeight, na);

},

scrollToPosition: function(na, oa) {

this.\_scrollTopAnimation && this.\_scrollTopAnimation.stop();

if (oa === true) {

this.\_scrollTopAnimation = (new g(this.scrollContainer)).to('scrollTop', na).ease(g.ease.end).duration(400).go();

} else

this.scrollContainer.scrollTop = na;

},

\_onStickerClick: function(na, oa) {

if (!na)

return;

j.log('StickersLoggerConfig', {

event: 'click\_sticker',

packid: na,

stickerid: oa

});

var pa = this.tabView;

z.onTrayDataReady(function() {

if (ea(z.getPackIDsInTray(), na)) {

var qa = z.getPack(na);

if (qa && qa.isPromoted)

y.addPack(na);

pa.setStickersFlyoutPackID(na);

} else

aa.showStore(na);

});

},

\_checkToAnimateSticker: function() {

var na = ga(this.scrollContainer);

h.inform('chatScrolled/' + this.threadID, {

scrollTop: this.scrollContainer.scrollTop,

top: na.y,

viewHeight: na.height

});

},

setNewThreadID: function(na) {

this.\_store.setNewThreadID(na);

}

});

e.exports = ma;

}, null);

\_\_d("ChatTabTypeaheadRenderer", ["MercuryConstants", "DOM", "CompactTypeaheadRenderer", "MercuryThreadMetadataRawRenderer", "TypeaheadFacepile"], function(a, b, c, d, e, f, g, h, i, j, k) {

function l(n, o) {

switch (n.type) {

case g.MercuryTypeaheadConstants.USER\_TYPE:

case g.MercuryTypeaheadConstants.PAGE\_TYPE:

return i(n, o);

case g.MercuryTypeaheadConstants.THREAD\_TYPE:

return m(n, o);

default:

return h.create('div');

}

}

function m(n, o) {

var p = [], q = n.mercury\_thread, r = n.participants\_to\_render.map(function(u) {

return u.big\_image\_src || u.image\_src;

});

if (r) {

if (r instanceof Array) {

r = k.render(r);

} else

r = h.create('img', {

alt: '',

src: r

});

p.push(r);

}

var s = q.name, t = j.renderRawParticipantList(q.thread\_id, n.participants\_to\_render, q.participants.length, {

names\_renderer: j.renderShortNames

});

p.push(h.create('span', {

className: 'timestamp'

}, q.timestamp\_relative));

p.push(h.create('span', {

className: 'text'

}, s));

p.push(h.create('span', {

className: 'subtext'

}, t));

return h.create('li', {

className: 'node'

}, p);

}

l.className = 'compact';

e.exports = l;

}, null);

\_\_d("MercuryTypeahead", ["Event", "ArbiterMixin", "DOM", "DOMDimensions", "Input", "Keys", "MercuryTypeaheadTemplates", "Tokenizer", "Typeahead", "TypeaheadCore", "copyProperties", "cx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r) {

var s = function(t, u) {

this.\_domElement = null;

this.\_typeahead = null;

this.\_tokenizer = null;

this.\_placeholder = '';

this.\_exclusions = [];

this.\_viewNodeOrID = null;

this.\_viewOptions = {

renderer: 'compact',

autoSelect: true

};

this.\_tokenizerBehaviors = [];

this.\_heightPrev = null;

this.\_dataSource = t;

this.\_view = u;

};

q(s.prototype, h);

q(s.prototype, {

setPlaceholder: function(t) {

this.\_placeholder = t;

return this;

},

setExcludedParticipantsFromThreadMeta: function(t) {

if (!t)

return;

if (!t.former\_participants) {

this.setExcludedParticipants(t.participants);

return;

}

var u = t.former\_participants.filter(function(v) {

return v.is\_friend === false;

}).map(function(v) {

return v.id;

});

this.setExcludedParticipants(u.concat(t.participants));

},

setExcludedParticipants: function(t) {

this.\_exclusions = [];

t.forEach(function(u) {

var v = u.indexOf(':');

if (u.substr(0, v) == 'fbid')

this.\_exclusions.push(u.substr(v + 1));

}.bind(this));

return this;

},

setViewNodeID: function(t) {

this.\_viewNodeOrID = t;

},

setViewNode: function(t) {

this.\_viewNodeOrID = t;

},

setFullWidthView: function(t) {

var u = i.create('div', {

className: "\_4ck uiTypeaheadView"

});

i.setContent(t, u);

this.setViewNode(u);

},

setViewOption: function(t, u) {

this.\_viewOptions[t] = u;

},

addTokenizerBehavior: function(t) {

this.\_tokenizerBehaviors.push(t);

},

build: function(t) {

if (this.\_domElement)

return;

var u = m[':fb:mercury:tokenizer'].build(), v = m[':fb:mercury:typeahead'].build();

this.\_domElement = u.getRoot();

i.appendContent(this.\_domElement, v.getRoot());

var w = v.getNode('textfield');

k.setPlaceholder(w, this.\_placeholder);

w.setAttribute('data-placeholder', this.\_placeholder);

this.\_input = w;

var x = {

node\_id: this.\_viewNodeOrID,

ctor: this.\_view,

options: this.\_viewOptions

}, y = {

ctor: p,

options: {

setValueOnSelect: true

}

};

this.\_typeahead = new o(this.\_dataSource, x, y, v.getRoot());

this.\_typeahead.init();

var z = {

inline: true,

behaviors: this.\_tokenizerBehaviors

};

this.\_tokenizer = new n(this.\_domElement, this.\_typeahead);

this.\_tokenizer.init(u.getNode('tokenarea'), 'participants', [], z);

this.\_tokenizer.subscribe(['addToken', 'removeToken', 'removeAllTokens'], this.\_tokensChanged.bind(this));

this.\_tokenizer.subscribe('resize', function() {

this.inform('resize');

}.bind(this));

g.listen(w, 'focus', function() {

this.\_resetDataSource();

this.\_typeahead.init();

}.bind(this));

g.listen(this.\_domElement, 'click', this.focus.bind(this));

g.listen(w, 'keydown', this.keydown.bind(this));

this.\_heightPrev = j.getElementDimensions(this.\_domElement).height;

},

getElement: function() {

return this.\_domElement;

},

getSelectedParticipantIDs: function() {

var t = [];

if (this.\_tokenizer)(this.\_tokenizer.getTokenValues() || []).forEach(function(u) {

t.push('fbid:' + u);

});

return t;

},

getTokens: function() {

var t = [];

if (this.\_tokenizer)

t = this.\_tokenizer.getTokens();

return t;

},

getTokenizer: function() {

return this.\_tokenizer;

},

keydown: function(event) {

if (this.\_tokenizer.inline && event.keyCode == l.ESC) {

if (k.isEmpty(this.\_input)) {

var t = this.\_tokenizer.getLastToken();

if (t && t.isRemovable())

this.\_tokenizer.removeToken(t);

} else

this.\_typeahead.getCore().reset();

return false;

}

if (k.isEmpty(this.\_input) && this.\_tokenizer.inline && event.keyCode === l.RETURN) {

event.preventDefault();

return this.inform('tokens-return');

}

},

reset: function() {

this.\_tokenizer && this.\_tokenizer.removeAllTokens();

this.\_typeahead && this.\_typeahead.getCore().reset();

},

focus: function() {

this.\_tokenizer && this.\_tokenizer.focusInput();

},

getTypeahead: function() {

return this.\_typeahead;

},

\_resetDataSource: function() {

this.\_dataSource.setExclusions(this.\_exclusions);

},

\_tokensChanged: function() {

this.inform('tokens-changed');

}

});

e.exports = s;

}, null);

\_\_d("ChatAddFriendsTabSheetRawRenderer", ["ChatTabTypeaheadRenderer", "ContextualTypeaheadView", "DOM", "Event", "MercuryConfig", "MercuryDataSourceWrapper", "MercuryTypeahead", "ChatTabTemplates", "fbt", "tx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

var q = {

render: function(r, s, t, u, v, w) {

var x = w ? n[':fb:mercury:chat:tab-sheet:add-friends-empty-tab'].build(): n[':fb:mercury:chat:tab-sheet:add-friends'].build(), y = {};

if (w) {

if (k.GroupThreadNewMessageGK) {

y = l.chat\_tab\_composer\_source;

y.setShowThreads(true);

} else

y = l.source;

} else

y = l.add\_people\_source;

var z = new m(y, h);

if (k.GroupThreadNewMessageGK && w && r != null)

z.setViewOption('renderer', g);

z.setExcludedParticipantsFromThreadMeta(u);

if (!w)

z.setPlaceholder("Freunde zu diesem Chat hinzuf\u00fcgen");

z.build();

i.replace(x.getNode('participantsTypeahead'), z.getElement());

i.setContent(t, x.getRoot());

z.getTokenizer().adjustWidth();

var aa = x.getNode('participantsTypeahead').getAttribute('data-labelid'), ba = i.scry(z.getElement(), 'input[type="text"]')[0];

if (ba)

if (aa) {

ba.setAttribute('aria-labelledby', aa);

} else

ba.setAttribute('aria-label', "An");

z.focus();

if (!w) {

var ca = function() {

var da = z.getSelectedParticipantIDs();

if (da.length)

v(da);

s.close(r);

};

j.listen(x.getNode('doneButton'), 'click', ca);

z.subscribe('tokens-return', function() {

if (z.getTokens().length)

ca();

});

}

return z;

}

};

e.exports = q;

}, null);

\_\_d("MultiChatController", ["AsyncSignal", "ChatOpenTab", "copyProperties", "Form", "MercuryIDs", "MercuryLocalIDs", "MercuryServerRequests", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = b('MercuryServerRequests').get(), n = b('MercuryThreads').get();

function o() {}

i(o, {

subscribe: function(p, q) {

p.subscribe('confirm', function() {

this.createGroupThreadFromChooserDialog(p, q);

}.bind(this));

},

createGroupThreadFromChooserDialog: function(p, q) {

var r = j.serialize(p.getRoot()), s = JSON.parse(r.profileChooserItems), t = [];

for (var u in s)

if (s[u])

t.push(u);

var v = o.createThreadForFBIDs(t);

m.subscribe('update-thread-ids', function(w, x) {

for (var y in x)

if (x[y] == v)

new g('/ajax/groups/chat/log', {

group\_id: q,

message\_id: y

}).send();

});

p.hide();

},

createThreadForTokens: function(p) {

if (!p.length)

return;

var q = p.length === 1 ? k.getThreadIDFromUserID(p[0].split(':')[1]): l.generateThreadID();

n.createNewLocalThread(q, p);

h.openThread(q, 'MultiChatController');

},

createThreadForFBIDs: function(p) {

var q = [];

for (var r = 0; r < p.length; r++)

q.push("fbid:" + p[r]);

return o.createThreadForTokens(q);

}

});

e.exports = o;

}, null);

\_\_d("ChatAddFriendsTabSheet", ["Arbiter", "ArbiterMixin", "ChatAddFriendsTabSheetRawRenderer", "ChatOpenTab", "MercuryConfig", "MercuryConstants", "MercuryLogMessageType", "MercurySourceType", "MultiChatController", "Style", "copyProperties", "MercuryMessages", "MercuryThreadActions", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q) {

var r = b('MercuryMessages').get(), s = b('MercuryThreadActions').get(), t = b('MercuryThreads').get();

function u(y, z, aa) {

this.\_threadID = y;

this.\_rootElement = z;

this.\_sheetView = aa;

this.\_typeahead = null;

}

q(u.prototype, h, {

render: function() {

t.getThreadMeta(this.\_threadID, function(y) {

var z = y.is\_canonical\_user ? v: w, aa = t.isNewEmptyLocalThread(this.\_threadID);

this.\_typeahead = i.render(this, this.\_sheetView, this.\_rootElement, y, z.bind(null, y), aa);

this.\_typeahead.subscribe('tokens-changed', function() {

if (k.GroupThreadNewMessageGK) {

if (aa)

x(this, this.\_typeahead, this.\_threadID);

} else

this.inform('chat/tokens-changed', {

token\_count: this.\_typeahead.getTokens().length

});

}.bind(this));

this.\_typeahead.subscribe('resize', function() {

this.\_sheetView.resize();

}.bind(this));

}.bind(this));

},

getParticipants: function() {

if (!this.\_typeahead)

return null;

return this.\_typeahead.getSelectedParticipantIDs();

},

isPermanent: function() {

return true;

},

isSheetWithInput: function() {

return true;

},

getType: function() {

return 'add\_friends\_type';

},

open: function() {

this.\_sheetView.open(this);

},

close: function() {

this.\_sheetView.close(this);

},

getHeight: function() {

return p.get(this.\_rootElement, 'height');

}

});

function v(y, z) {

var aa = y.participants;

o.createThreadForTokens(aa.concat(z));

}

function w(y, z) {

var aa = y.thread\_id;

if (t.isEmptyLocalThread(aa)) {

s.addParticipants(aa, z);

} else

r.sendMessage(r.constructLogMessageObject(n.CHAT\_WEB, aa, m.SUBSCRIBE, {

added\_participants: z

}));

j.openThread(aa, 'ChatAddFriendsTabSheet');

}

function x(y, z, aa) {

var ba = z.getTypeahead().getData(), ca = z.getTokens();

ba.setShowThreads(ca.length === 0);

var da = ca.length == 1 && ca[0].getInfo().render\_type === l.MercuryTypeaheadConstants.THREAD\_TYPE;

if (da) {

var ea = ca[0].getInfo().mercury\_thread;

if (ea)

j.openThread(ea.thread\_fbid, 'openGroupChat', '');

g.inform('chat/close-tab', aa, 'open\_group\_chat');

} else

y.inform('chat/tokens-changed', {

token\_count: z.getTokens().length

});

}

e.exports = u;

}, null);

\_\_d("ChatNameConversationTabSheetRawRenderer", ["DOM", "Event", "Input", "Keys", "ChatTabTemplates", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

"use strict";

var m = {

render: function(n, o, p, q, r, s) {

var t = k[':fb:mercury:chat:tab-sheet:name-conversation'].build(), u = t.getNode('nameInput'), v = t.getNode('doneButton'), w = "Fertig", x = "Verbergen", y = q.name;

if (y) {

i.setValue(u, y);

} else

g.setContent(v, x);

var z = function() {

var aa = i.getValue(u);

if (aa)

r(aa);

s();

o.close(n);

};

h.listen(u, 'input', function() {

g.setContent(v, i.getValue(u).length === 0 ? x : w);

});

h.listen(v, 'click', z);

h.listen(u, 'keyup', function(aa) {

if (aa.keyCode === j.RETURN) {

z();

return false;

}

});

g.setContent(p, t.getRoot());

!n.isAutomatic() && u.focus();

}

};

e.exports = m;

}, null);

\_\_d("ChatNameConversationTabSheet", ["AsyncRequest", "ChatNameConversationTabSheetRawRenderer", "MercuryAPIArgsSource", "MercuryLogMessageType", "MercurySourceType", "MercuryMessages", "MercuryServerRequests", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = b('MercuryMessages').get(), m = b('MercuryServerRequests').get(), n = b('MercuryThreads').get(), o = '/ajax/chat/multichat/name\_conversation/dismiss.php', p = {};

function q(r, s, t) {

"use strict";

this.$ChatNameConversationTabSheet0 = r;

this.$ChatNameConversationTabSheet1 = s;

this.$ChatNameConversationTabSheet2 = t;

this.$ChatNameConversationTabSheet3 = false;

}

q.prototype.render = function() {

"use strict";

n.getThreadMeta(this.$ChatNameConversationTabSheet0, function(r) {

h.render(this, this.$ChatNameConversationTabSheet2, this.$ChatNameConversationTabSheet1, r, this.$ChatNameConversationTabSheet4.bind(null, r), this.$ChatNameConversationTabSheet5.bind(this, r));

this.$ChatNameConversationTabSheet2.resize();

}.bind(this));

};

q.prototype.isPermanent = function() {

"use strict";

return true;

};

q.prototype.isSheetWithInput = function() {

"use strict";

return true;

};

q.prototype.getType = function() {

"use strict";

return 'name\_conversation\_type';

};

q.prototype.open = function(r) {

"use strict";

this.$ChatNameConversationTabSheet3 = r;

if (!(this.$ChatNameConversationTabSheet3 && this.$ChatNameConversationTabSheet6()))

this.$ChatNameConversationTabSheet2.open(this);

};

q.prototype.close = function() {

"use strict";

this.$ChatNameConversationTabSheet2.close(this);

};

q.prototype.isAutomatic = function() {

"use strict";

return this.$ChatNameConversationTabSheet3;

};

q.prototype.$ChatNameConversationTabSheet6 = function() {

"use strict";

return !!p[this.$ChatNameConversationTabSheet0];

};

q.prototype.$ChatNameConversationTabSheet7 = function() {

"use strict";

p[this.$ChatNameConversationTabSheet0] = true;

};

q.prototype.$ChatNameConversationTabSheet5 = function(r) {

"use strict";

if (!r.name\_conversation\_sheet\_dismissed ||!this.$ChatNameConversationTabSheet6()) {

this.$ChatNameConversationTabSheet7();

m.getServerThreadID(r.thread\_id, function(s) {

new g(o).setData({

thread\_id: s

}).send();

});

}

};

q.prototype.$ChatNameConversationTabSheet4 = function(r, s) {

"use strict";

var t = r.name;

if ((s || t) && (s != t))

l.sendMessage(l.constructLogMessageObject(k.CHAT\_WEB, r.thread\_id, j.THREAD\_NAME, {

name: s

}), null, i.CHAT);

};

e.exports = q;

}, null);

\_\_d("ChatNewMessagesTabSheet", ["Event", "ArbiterMixin", "DOM", "ChatTabTemplates", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k) {

function l(m, n, o) {

this.\_threadID = m;

this.\_rootElement = n;

this.\_sheetView = o;

}

k(l.prototype, h, {

render: function() {

var m = j[':fb:mercury:chat:tab-sheet:clickable-message-icon-sheet'].build();

i.setContent(m.getNode('text'), i.tx.\_("Scrolle nach unten, um neue Nachrichten zu sehen."));

i.setContent(this.\_rootElement, m.getRoot());

g.listen(m.getRoot(), 'click', function() {

this.inform('clicked', this.\_threadID);

}.bind(this));

},

isPermanent: function() {

return true;

},

isSheetWithInput: function() {

return true;

},

getType: function() {

return 'new\_messages\_type';

},

open: function() {

this.\_sheetView.open(this);

},

close: function() {

this.\_sheetView.close(this);

}

});

e.exports = l;

}, null);

\_\_d("ChatNoRecipientsTabSheet", ["DOM", "MercuryViewer", "ChatTabTemplates", "copyProperties", "fbt", "MercuryThreads", "MercuryThreadInformer"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = b('MercuryThreads').get(), m = b('MercuryThreadInformer').get();

function n(o, p, q) {

this.\_threadID = o;

this.\_rootElement = p;

this.\_sheetView = q;

m.subscribe('threads-updated', this.\_handleThreadsUpdated.bind(this));

}

j(n.prototype, {

render: function() {

var o = i[':fb:mercury:chat:tab-sheet:message-icon-sheet'].build();

g.setContent(o.getNode('text'), "Alle Teilnehmer haben die Unterhaltung verlassen.");

g.setContent(this.\_rootElement, o.getRoot());

},

isPermanent: function() {

return true;

},

getType: function() {

return 'no\_recipients\_type';

},

\_handleThreadsUpdated: function() {

l.getThreadMeta(this.\_threadID, function(o) {

var p = h.getID(), q = o.participants.filter(function(r) {

return r != p;

});

if (q.length < 1&&!l.isNewEmptyLocalThread(this.\_threadID)) {

this.\_sheetView.open(this);

} else

this.\_sheetView.close(this);

}.bind(this));

}

});

e.exports = n;

}, null);

\_\_d("ChatOfflineTabSheet", ["ChatPrivacyActionController", "ChatVisibility", "CSS", "DOM", "Event", "JSLogger", "MercuryIDs", "MercuryParticipants", "ChatTabTemplates", "copyProperties", "cx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q) {

function r(s, t, u) {

this.\_rootElement = t;

this.\_sheetView = u;

this.\_logger = l.create('blackbird');

this.\_canonicalUser = m.getUserIDFromThreadID(s);

if (this.\_canonicalUser)

this.\_privacyActionController = new g(this.\_canonicalUser, this.\_handlePrivacyChange.bind(this));

}

p(r.prototype, {

render: function() {

if (!this.\_canonicalUser) {

this.\_logger.error('offline\_sheet\_open\_with\_non\_friend');

return;

}

var s = o[':fb:mercury:chat:tab-sheet:message-icon-sheet'].build(), t = 'fbid:' + this.\_canonicalUser;

n.get(t, function(u) {

var v = 'fbChatGoOnlineLink', w = j.tx.\_("aktiviere den Chat"), x = j.create('a', {

href: '#',

className: v

}, w), y = j.tx.\_("Um mit {name} und weiteren Freunden zu chatten, {link}.", {

name: u.short\_name,

link: x

});

j.setContent(s.getNode('text'), y);

i.addClass(s.getRoot(), "\_1sk1");

j.setContent(this.\_rootElement, s.getRoot());

k.listen(this.\_rootElement, 'click', function(event) {

if (i.hasClass(event.getTarget(), v)) {

if (h.isOnline())

this.\_logger.error('tab\_sheet\_already\_online');

this.\_privacyActionController.togglePrivacy();

this.\_logger.log('tab\_sheet\_go\_online', {

tab\_id: this.\_canonicalUser

});

return false;

}

}.bind(this));

this.\_sheetView.resize();

}.bind(this));

},

\_handlePrivacyChange: function(s) {

if (!this.\_canonicalUser)

this.\_logger.error('user\_blocked\_sheet\_privacy\_changed\_non\_friend');

switch (s) {

case g.OFFLINE:

this.\_sheetView.open(this);

break;

case g.NORMAL:

case g.BLOCKED:

this.\_sheetView.close(this);

break;

}

},

isPermanent: function() {

return true;

},

getType: function() {

return 'offline\_type';

},

destroy: function() {

this.\_privacyActionController && this.\_privacyActionController.destroy();

}

});

e.exports = r;

}, null);

\_\_d("ChatUploadWarningTabSheet", ["DOM", "ChatTabTemplates", "copyProperties"], function(a, b, c, d, e, f, g, h, i) {

function j(k, l, m) {

this.\_threadID = k;

this.\_rootElement = l;

this.\_sheetView = m;

}

i(j.prototype, {

render: function() {

var k = h[':fb:mercury:chat:tab-sheet:message-icon-sheet'].build();

g.setContent(k.getNode('text'), g.tx.\_("Bitte warte mit dem Senden deiner Nachricht bis das Hochladen abgeschlossen ist."));

g.setContent(this.\_rootElement, k.getRoot());

},

isPermanent: function() {

return false;

},

getType: function() {

return 'upload\_warning\_type';

},

open: function() {

this.\_sheetView.open(this);

},

close: function() {

this.\_sheetView.close(this);

}

});

e.exports = j;

}, null);

\_\_d("MercuryReadOnlyReason", ["React", "fbt"], function(a, b, c, d, e, f, g, h) {

var i = {

getReason: function(j) {

if (j.has\_email\_participant) {

return (h.\_("Leider ist diese Unterhaltung nicht mehr aktiv, weil Nachrichten nicht mehr an E-Mail-Adressen gesendet werden k\u00f6nnen. Nachrichten an Facebook.com-E-Mail-Adressen werden jetzt an prim\u00e4re E-Mail-Adressen weitergeleitet, damit du diese Unterhaltung per E-Mail fortsetzen kannst. {Learn more}.", [h.param("Learn more", g.createElement("a", {

href: "/help/224049364288051"

}, "Mehr dazu"))]));

} else

return ("Du kannst nicht auf diese Unterhaltung antworten.");

}

};

e.exports = i;

}, null);

\_\_d("ChatReadOnlyTabSheet", ["React", "MercuryReadOnlyReason", "cx", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i) {

var j = b('MercuryThreads').get();

function k(l, m, n) {

"use strict";

this.$ChatReadOnlyTabSheet0 = l;

this.$ChatReadOnlyTabSheet1 = m;

this.$ChatReadOnlyTabSheet2 = n;

}

k.prototype.render = function() {

"use strict";

var l = h.getReason(j.getThreadMetaNow(this.$ChatReadOnlyTabSheet0));

g.render(g.createElement("div", {

className: "\_87-"

}, l), this.$ChatReadOnlyTabSheet1);

};

k.prototype.isPermanent = function() {

"use strict";

return false;

};

k.prototype.getType = function() {

"use strict";

return 'chat-thread-is-read-only';

};

k.prototype.open = function() {

"use strict";

this.$ChatReadOnlyTabSheet2.open(this);

};

k.prototype.close = function() {

"use strict";

this.$ChatReadOnlyTabSheet2.open(this);

};

e.exports = k;

}, null);

\_\_d("ChatThreadIsMutedTabSheet", ["Event", "ArbiterMixin", "DOM", "ChatTabTemplates", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k) {

function l(m, n, o) {

this.\_threadID = m;

this.\_rootElement = n;

this.\_sheetView = o;

}

k(l.prototype, h, {

render: function() {

var m = j[':fb:mercury:chat:tab-sheet:message-mute-sheet'].build();

i.setContent(m.getNode('text'), i.tx.\_("Der Ton ist f\u00fcr diese Unterhaltung ausgeschaltet. Chat-Reiter werden nicht angezeigt und Pushbenachrichtigungen sind deaktiviert."));

i.setContent(this.\_rootElement, m.getRoot());

g.listen(m.getNode('unmuteButton'), 'click', function() {

this.inform('clicked', this.\_threadID);

}.bind(this));

},

isPermanent: function() {

return false;

},

getType: function() {

return 'chat-thread-is-muted';

},

open: function() {

this.\_sheetView.open(this);

},

close: function() {

this.\_sheetView.close(this);

}

});

e.exports = l;

}, null);

\_\_d("ChatEmployeeAwaySheet", ["ArbiterMixin", "DOM", "mixin", "ChatTabTemplates"], function(a, b, c, d, e, f, g, h, i, j) {

var k = i(g);

for (var l in k)

if (k.hasOwnProperty(l))

n[l] = k[l];

var m = k === null ? null: k.prototype;

n.prototype = Object.create(m);

n.prototype.constructor = n;

n.\_\_superConstructor\_\_ = k;

function n(o, p, q) {

"use strict";

this.$ChatEmployeeAwaySheet0 = o;

this.$ChatEmployeeAwaySheet1 = p;

this.$ChatEmployeeAwaySheet2 = q;

}

n.prototype.render = function() {

"use strict";

var o = j[':fb:mercury:employee-away-sheet'].build();

h.setContent(this.$ChatEmployeeAwaySheet1, o.getRoot());

};

n.prototype.isPermanent = function() {

"use strict";

return true;

};

n.prototype.getType = function() {

"use strict";

return 'employee-away';

};

n.prototype.open = function() {

"use strict";

this.$ChatEmployeeAwaySheet2.open(this);

};

n.prototype.close = function() {

"use strict";

this.$ChatEmployeeAwaySheet2.close(this);

};

e.exports = n;

}, null);

\_\_d("ChatUserBlockedTabSheet", ["CSS", "ChatPrivacyActionController", "DOM", "Event", "GenderConst", "JSLogger", "MercuryIDs", "MercuryParticipants", "ChatTabTemplates", "copyProperties", "cx", "tx"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r) {

function s(t, u, v) {

this.\_rootElement = u;

this.\_sheetView = v;

this.\_logger = l.create('blackbird');

this.\_canonicalUser = m.getUserIDFromThreadID(t);

if (this.\_canonicalUser)

this.\_privacyActionController = new h(this.\_canonicalUser, this.\_handlePrivacyChange.bind(this));

}

p(s.prototype, {

render: function() {

if (!this.\_canonicalUser) {

this.\_logger.error('user\_blocked\_sheet\_open\_with\_non\_friend');

return;

}

var t = o[':fb:mercury:chat:tab-sheet:user-blocked'].build(), u = 'fbid:' + this.\_canonicalUser;

n.get(u, function(v) {

var w = null;

switch (v.gender) {

case k.FEMALE\_SINGULAR:

case k.FEMALE\_SINGULAR\_GUESS:

w = r.\_("Du hast den Chat f\u00fcr {name} ausgeschaltet. Du kannst ihr aber trotzdem eine Nachricht schicken.", {

name: v.short\_name

});

break;

case k.MALE\_SINGULAR:

case k.MALE\_SINGULAR\_GUESS:

w = r.\_("Du hast den Chat f\u00fcr {name} ausgeschaltet. Du kannst ihm aber trotzdem eine Nachricht schicken.", {

name: v.short\_name

});

break;

default:

w = r.\_("Du hast den Chat f\u00fcr {name} ausgeschaltet. Du kannst ihm\/ihr aber trotzdem eine Nachricht schicken.", {

name: v.short\_name

});

}

i.setContent(t.getNode('text'), w);

var x = r.\_("Chat f\u00fcr {name} einschalten?", {

name: v.short\_name

});

i.setContent(t.getNode('actionLink'), x);

g.addClass(t.getRoot(), "\_1sk0");

i.setContent(this.\_rootElement, t.getRoot());

j.listen(t.getNode('actionLink'), 'click', this.\_privacyActionController.togglePrivacy.bind(this.\_privacyActionController));

this.\_sheetView.resize();

}.bind(this));

},

\_handlePrivacyChange: function(t) {

if (!this.\_canonicalUser)

this.\_logger.error('user\_blocked\_sheet\_privacy\_changed\_non\_friend');

if (t == h.BLOCKED) {

var u = 'fbid:' + this.\_canonicalUser;

n.get(u, function(v) {

if (v.is\_friend) {

this.\_sheetView.open(this);

} else

this.\_sheetView.close(this);

}.bind(this));

return;

}

this.\_sheetView.close(this);

},

isPermanent: function() {

return true;

},

getType: function() {

return 'user\_blocked\_type';

},

destroy: function() {

this.\_privacyActionController && this.\_privacyActionController.destroy();

}

});

e.exports = s;

}, null);

\_\_d("MercurySheetPolicy", [], function(a, b, c, d, e, f) {

var g = {

canReplaceOpenSheet: function(h, i) {

if (h.isSheetWithInput && h.isSheetWithInput())

return i.getType() != h.getType() && i.isSheetWithInput && i.isSheetWithInput();

if (h.getType() == i.getType())

return false;

if (h.isPermanent()&&!i.isPermanent())

return false;

return true;

}

};

e.exports = g;

}, null);

\_\_d("MercurySheetView", ["Animation", "ArbiterMixin", "MercurySheetPolicy", "CSS", "DOM", "Style", "MercurySheetTemplates", "Vector", "copyProperties", "cx", "setTimeoutAcrossTransitions"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q) {

var r = 5000, s = function(t, u, v) {

this.\_threadID = t;

this.\_rootElement = u;

this.\_tabMainElement = v;

this.\_openSheet = null;

};

o(s.prototype, h, {

destroy: function() {

k.empty(this.\_rootElement);

},

\_openCommon: function(t, u) {

if (this.\_openSheet&&!i.canReplaceOpenSheet(this.\_openSheet, t)) {

if (t.couldNotReplace)

t.couldNotReplace();

return;

}

this.clear(function() {

this.\_openSheet = t;

var v = m[':fb:mercury:tab-sheet:loading'].build().getRoot();

k.setContent(this.\_rootElement, v);

j.show(v);

j.show(this.\_rootElement);

t.render();

if (u) {

j.addClass(this.\_tabMainElement, 'sheetSlide');

j.addClass(this.\_tabMainElement, "\_1sk4");

var w = n.getElementDimensions(this.\_rootElement).y;

l.set(this.\_rootElement, 'bottom', w + 'px');

this.resize();

this.\_animation = new g(this.\_rootElement).to('bottom', 0).duration(150).ease(g.ease.both).ondone(function() {

j.removeClass(this.\_tabMainElement, 'sheetSlide');

j.removeClass(this.\_tabMainElement, "\_1sk4");

this.resize();

}.bind(this)).go();

} else

this.resize();

if (!t.isPermanent()) {

var x = r;

if (t.getCloseTimeout)

x = t.getCloseTimeout();

var y = this.getAutoCloseCallback(t);

this.\_sheetCloseHandler = q(this.close.bind(this, t, y), x);

if (t.timeoutCanBeReset)

t.setResetTimeoutCallback(this.resetTimeout.bind(this));

}

}.bind(this));

},

getAutoCloseCallback: function(t) {

if (!t.autoCloseCallback)

return null;

return t.autoCloseCallback.bind(t);

},

resetTimeout: function(t, u) {

clearTimeout(this.\_sheetCloseHandler);

var v = this.getAutoCloseCallback(t);

this.\_sheetCloseHandler = q(this.close.bind(this, t, v), u);

},

set: function(t) {

return this.\_openCommon(t, false);

},

open: function(t) {

return this.\_openCommon(t, true);

},

close: function(t, u) {

if (this.\_openSheet != t)

return;

if (!this.\_openSheet) {

u && u();

return;

}

if (this.\_animation)

this.\_animation.stop();

if (this.\_sheetCloseHandler) {

clearTimeout(this.\_sheetCloseHandler);

this.\_sheetCloseHandler = null;

}

j.addClass(this.\_tabMainElement, 'sheetSlide');

j.addClass(this.\_tabMainElement, "\_1sk4");

var v = n.getElementDimensions(this.\_rootElement).y;

this.resize();

this.\_animation = new g(this.\_rootElement).to('bottom', v + 'px').duration(100).ease(g.ease.begin).ondone(function() {

k.empty(this.\_rootElement);

j.hide(this.\_rootElement);

j.removeClass(this.\_tabMainElement, 'sheetSlide');

j.removeClass(this.\_tabMainElement, "\_1sk4");

this.\_openSheet = null;

this.resize();

u && u();

}.bind(this)).go();

},

clear: function(t) {

this.close(this.\_openSheet, t);

},

resize: function() {

this.inform('resize');

}

});

e.exports = s;

}, null);

\_\_d("ChatTabSheetController", ["ChatAddFriendsTabSheet", "ChatNameConversationTabSheet", "ChatNewMessagesTabSheet", "ChatNoRecipientsTabSheet", "ChatOfflineTabSheet", "ChatUploadWarningTabSheet", "ChatReadOnlyTabSheet", "ChatThreadIsMutedTabSheet", "ChatEmployeeAwaySheet", "ChatUserBlockedTabSheet", "MercuryIDs", "MercurySheetView", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s) {

var t = function(u, v, w) {

this.\_sheetView = new r(u, v, w);

this.\_addFriendsTabSheet = new g(u, v, this.\_sheetView);

this.\_nameConversationTabSheet = new h(u, v, this.\_sheetView);

this.\_userBlockedTabSheet = new p(u, v, this.\_sheetView);

this.\_offlineTabSheet = new k(u, v, this.\_sheetView);

this.\_newMessagesTabSheet = new i(u, v, this.\_sheetView);

this.\_uploadWarningTabSheet = new l(u, v, this.\_sheetView);

this.\_threadIsMutedTabSheet = new n(u, v, this.\_sheetView);

this.\_employeeAwaySheet = new o(u, v, this.\_sheetView);

this.\_readOnlyTabSheet = new m(u, v, this.\_sheetView);

if (!q.getUserIDFromThreadID(u))

this.\_noRecipientsTabSheet = new j(u, v, this.\_sheetView);

};

s(t.prototype, {

openAddFriendsSheet: function() {

this.\_addFriendsTabSheet.open();

},

getAddFriendsTabSheet: function() {

return this.\_addFriendsTabSheet;

},

getAddFriendsParticipants: function() {

var u = this.\_addFriendsTabSheet.getParticipants();

this.\_addFriendsTabSheet.close();

return u;

},

openNameConversationSheet: function(u) {

this.\_nameConversationTabSheet.open(u);

},

openReadOnlySheet: function() {

this.\_readOnlyTabSheet.open();

},

openNewMessagesSheet: function() {

this.\_newMessagesTabSheet.open();

},

openUploadWarningTabSheet: function() {

this.\_uploadWarningTabSheet.open();

},

openThreadIsMutedTabSheet: function() {

this.\_threadIsMutedTabSheet.open();

},

openEmployeeAwaySheet: function() {

this.\_employeeAwaySheet.open();

},

closeAutomaticNameConversationSheet: function() {

if (this.\_nameConversationTabSheet.isAutomatic())

this.\_nameConversationTabSheet.close();

},

closeThreadIsMutedTabSheet: function() {

this.\_threadIsMutedTabSheet.close();

},

closeNewMessagesSheet: function() {

this.\_newMessagesTabSheet.close();

},

closeUploadWarningTabSheet: function() {

this.\_uploadWarningTabSheet.close();

},

onClickNewMessagesSheet: function(u) {

this.\_newMessageClickSub = this.\_newMessagesTabSheet.subscribe('clicked', u);

},

onClickThreadIsMutedSheet: function(u) {

this.\_threadIsMutedClickSub = this.\_threadIsMutedTabSheet.subscribe('clicked', u);

},

onResize: function(u) {

this.\_sheetView.subscribe('resize', u);

},

onTokensChanged: function(u) {

this.\_addFriendsTabSheet.subscribe('chat/tokens-changed', u);

},

destroy: function() {

this.\_sheetView && this.\_sheetView.destroy();

this.\_offlineTabSheet && this.\_offlineTabSheet.destroy();

this.\_userBlockedTabSheet && this.\_userBlockedTabSheet.destroy();

this.\_newMessageClickSub && this.\_newMessageClickSub.unsubscribe();

this.\_threadIsMutedClickSub && this.\_threadIsMutedClickSub.unsubscribe();

}

});

e.exports = t;

}, null);

\_\_d("ChatTitleLink.react", ["Link.react", "MercuryThreadTitle.react", "React", "TrackingNodes", "cx"], function(a, b, c, d, e, f, g, h, i, j, k) {

"use strict";

var l = i.createClass({

displayName: 'ChatTitleLink',

propTypes: {

href: i.PropTypes.string,

isNewThread: i.PropTypes.bool,

thread: i.PropTypes.object,

viewer: i.PropTypes.string

},

getDefaultProps: function() {

return {

href: null

};

},

getTitle: function() {

return this.refs.threadTitle.getTitle();

},

render: function() {

if (!this.props.thread)

return null;

return (i.createElement(g, {

'aria-level': "3",

className: (("titlebarText") + (this.props.href === null ? ' ' + "noLink" : '')),

'data-ft': j.getTrackingInfo(j.types.TITLE),

'data-hover': "tooltip",

href: this.props.href

}, i.createElement(h, {

ref: "threadTitle",

isNewThread: this.props.isNewThread,

thread: this.props.thread,

useAndSeparator: true,

useShortName: !this.props.thread.is\_canonical,

viewer: this.props.viewer

})));

}

});

e.exports = l;

}, null);

\_\_d("ChatVideoCallButton.react", ["Link.react", "ReactComponentWithPureRenderMixin", "React", "TrackingNodes", "cx", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

"use strict";

var m = i.createClass({

displayName: 'ChatVideoCallButton',

mixins: [h],

propTypes: {

onClick: i.PropTypes.func,

onKeyUp: i.PropTypes.func,

shown: i.PropTypes.bool,

tooltipLabel: i.PropTypes.string

},

getDefaultProps: function() {

return {

tooltipLabel: ("Videoanruf beginnen")

};

},

render: function() {

var n = j.getTrackingInfo(j.types.VIDEOCHAT), o = JSON.stringify({

videochat: 'call\_clicked\_bhat\_tab'

});

return (i.createElement(g, {

'aria-label': this.props.tooltipLabel,

className: (("videoicon") + (' ' + "button") + (!this.props.shown ? ' ' + "hidden\_elem" : '')),

'data-ft': n,

'data-gt': o,

'data-hover': "tooltip",

'data-tooltip-position': "above",

onClick: this.props.onClick,

onKeyUp: this.props.onKeyUp

}));

}

});

e.exports = m;

}, null);

\_\_d("EmoticonUtils", ["EmoticonsList", "Parent", "endsWith", "startsWith"], function(a, b, c, d, e, f, g, h, i, j) {

var k = 'emoticon\_', l = g.symbols, m = {

getEmoteFromTarget: function(n) {

var o = h.byClass(n, 'emoticon');

if (!o)

return null;

var p = null;

o.className.split(' ').forEach(function(q) {

if (j(q, k))

p = q.substring(k.length);

});

return l[p] || null;

},

insertEmoticon: function(n, o, p) {

var q = o.substring(0, p.start), r = o.substring(p.end);

if (p.start > 0&&!i(q, ' '))

n = ' ' + n;

if (!j(r, ' '))

n += ' ';

var s = q + n + r;

p.start += n.length;

p.end = p.start;

return {

result: s,

start: p.start,

end: p.end

};

}

};

e.exports = m;

}, null);

\_\_d("SelectionPosition", ["Event", "InputSelection", "SubscriptionsHandler", "merge"], function(a, b, c, d, e, f, g, h, i, j) {

function k(l) {

"use strict";

this.$SelectionPosition0 = l;

this.$SelectionPosition1 = {

start: 0,

end: 0

};

this.$SelectionPosition2 = new i();

this.$SelectionPosition3 = false;

this.updatePos();

this.enable();

}

k.prototype.getPos = function() {

"use strict";

return j(this.$SelectionPosition1);

};

k.prototype.setPos = function(l, m) {

"use strict";

this.$SelectionPosition1 = {

start: l,

end: m

};

this.restore();

};

k.prototype.updatePos = function() {

"use strict";

this.$SelectionPosition1 = h.get(this.$SelectionPosition0);

};

k.prototype.restore = function() {

"use strict";

h.set(this.$SelectionPosition0, this.$SelectionPosition1.start, this.$SelectionPosition1.end);

};

k.prototype.enable = function() {

"use strict";

if (!this.$SelectionPosition3) {

this.$SelectionPosition2.addSubscriptions(g.listen(this.$SelectionPosition0, 'keyup', this.updatePos.bind(this)), g.listen(this.$SelectionPosition0, 'click', this.updatePos.bind(this)));

this.$SelectionPosition3 = true;

}

};

k.prototype.disable = function() {

"use strict";

if (this.$SelectionPosition3) {

this.$SelectionPosition2.release();

this.$SelectionPosition3 = false;

}

};

e.exports = k;

}, null);

\_\_d("StickerSearchNUX.react", ["React", "ReactLayeredComponentMixin", "StickerServerRequests", "XUIAmbientNUX.react", "fbt"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = g.createClass({

displayName: 'StickerSearchNUX',

mixins: [h],

getInitialState: function() {

return {

nuxShown: true

};

},

componentDidMount: function() {

i.markSeenSearchTooltipNUX();

},

hideNUX: function() {

this.setState({

nuxShown: false

});

},

onCloseButtonClicked: function() {

this.hideNUX();

this.props.showIndicatorFunc();

},

renderLayers: function() {

return {

stickerSearchNUX: g.createElement(j, {

ref: "NUX",

contextRef: "container",

shown: this.state.nuxShown,

position: "above",

width: "auto",

onCloseButtonClick: this.onCloseButtonClicked

}, "Jetzt kannst du nach Stickern suchen!")

};

},

render: function() {

return (g.createElement("span", {

ref: "container"

}));

}

});

e.exports = l;

}, null);

\_\_d("XStickerReplyNUXControllerURIBuilder", ["XControllerURIBuilder"], function(a, b, c, d, e, f) {

e.exports = b("XControllerURIBuilder").create("\/stickers\/tooltip\/", {});

}, null);

\_\_d("ChatTabView", ["Arbiter", "ArbiterMixin", "AsyncDialog", "AsyncRequest", "AsyncSignal", "AvailableListConstants", "BanzaiODS", "ChatAddToThreadButton.react", "ChatArchiveWarning.react", "ChatBehavior", "ChatCloseButton.react", "ChatConfig", "ChatContextualDialogController", "ChatEmployeeAwayWarning", "ChatOpenTab", "ChatPhotoUploader.react", "ChatPrivacyActionController", "ChatQuietLinks", "ChatShareLinkUploader", "ChatStickerButton.react", "ChatStickerTriggerController", "ChatTabMenu.react", "ChatTabMessagesView", "ChatTabSheetController", "ChatTitleLink.react", "ChatVisibility", "ChatVideoCallButton.react", "MercuryConstants", "ContentRollFlyout.react", "CurrentUser", "CSS", "CurrentLocale", "Dialog", "Dock", "DOM", "DOMEvent", "DOMQuery", "DragDropTarget", "EmoticonsList", "EmoticonUtils", "Event", "Focus", "ImmutableObject", "Input", "Keys", "Locale", "MercuryConfig", "MercuryIDs", "MercuryFileUploader", "MercuryLogMessageType", "MercuryParticipants", "MercurySourceType", "MercuryThreadMetadataRawRenderer", "MercuryThreadMuter", "MercuryTypingAnimation.react", "MercuryTypingReceiver", "MercuryViewer", "NubController", "Parent", "PhotosUploadWaterfall", "PresencePrivacy", "PresenceStatus", "React", "Run", "SelectionPosition", "StickerActions", "StickerConfig", "StickerConstants", "StickersFlyout.react", "StickerState", "StickerSearchNUX.react", "Style", "SubscriptionsHandler", "SyntheticMouseEvent", "ChatTabTemplates", "TextAreaControl", "Toggler", "Tooltip", "TypingDetectorController", "URI", "UserAgent\_DEPRECATED", "VideoCallCore", "WaterfallIDGenerator", "WebMessengerThreadPermalinks", "XStickerReplyNUXControllerURIBuilder", "copyProperties", "cx", "csx", "fbt", "getActiveElement", "isEmpty", "setIntervalAcrossTransitions", "tx", "ChatReactInput", "OrionTrigger", "MercuryMessageActions", "MercuryMessages", "MercuryServerRequests", "MercuryThreadInformer", "MercuryThreadMetadataRenderer", "MercuryThreadActions", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, aa, ba, ca, da, ea, fa, ga, ha, ia, ja, ka, la, ma, na, oa, pa, qa, ra, sa, ta, ua, va, wa, xa, ya, za, ab, bb, cb, db, eb, fb, gb, hb, ib, jb, kb, lb, mb, nb, ob, pb, qb, rb, sb, tb, ub, vb, wb, xb, yb, zb, ac, bc, cc, dc, ec, fc, gc, hc, ic, jc, kc, lc, mc, nc, oc, pc, qc, rc, sc, tc, uc) {

"use strict";

var vc = b('ChatReactInput').module, wc = b('OrionTrigger').module, xc = b('MercuryMessageActions').get(), yc = b('MercuryMessages').get(), zc = b('MercuryServerRequests').get(), ad = b('MercuryThreadInformer').get(), bd = b('MercuryThreadMetadataRenderer').get(), cd = b('MercuryThreadActions').get(), dd = b('MercuryThreads').get(), ed = /^\s\*$/, fd = {}, gd = null, hd, id = 10;

function jd(ee, fe) {

var ge = oa.create('div');

fe = fe && fe.filter(function(je) {

return je != kb.getID();

});

if (sc(fe))

return fc.remove(ee);

var he = fe.length, ie = he - id;

if (ie > 1)

fe = fe.slice(0, id);

eb.getMulti(fe, function(je) {

for (var ke in je) {

var le = je[ke], me = cc[':fb:mercury:chat:multichat-tooltip-item'].build();

oa.setContent(me.getNode('name'), le.name);

var ne = bb.getUserIDFromParticipantID(ke), oe = ne && pb.get(ne) == l.ACTIVE;

ka.conditionShow(me.getNode('icon'), oe);

ka.conditionClass(me.getNode('name'), 'tooltipItemWithIcon', oe);

oa.appendContent(ge, me.getRoot());

}

if (ie > 1) {

var pe = qc.\_("und {number of hidden chat participants} weitere.", [qc.param("number of hidden chat participants", ie)]);

oa.appendContent(ge, oa.create('div', {}, pe));

}

fc.set(ee, ge, 'above');

});

}

var kd = {}, ld = null, md = false;

function nd(ee, fe, ge) {

if (ge) {

kd[ee] = fe;

if (!ld)

ld = tc(od, 600);

} else {

ka.removeClass(fe, 'highlightTitle');

delete kd[ee];

}

}

function od() {

for (var ee in kd) {

var fe = kd[ee];

if (fe.parentNode) {

ka.conditionClass(fe, 'highlightTitle', md);

} else

delete kd[ee];

}

md=!md;

if (!Object.keys(kd).length) {

clearInterval(ld);

ld = null;

}

}

function pd(ee) {

var fe = bb.tokenize(ee);

switch (fe.type) {

case 'user':

return cc[':fb:mercury:chat:user-tab'].build();

}

return cc[':fb:mercury:chat:multichat-tab'].build();

}

function qd(ee) {

var fe = ee.\_getInputValue(), ge = ee.\_fileUploader.getAttachments();

if (wd(ee))

sd(ee, fe, ge, function(he) {

var ie = ee.\_fileUploader.getImageFiles();

if (ie.length > 0) {

he.image\_ids = ie;

he.has\_attachment = true;

}

var je = ee.\_fileUploader.getVideoFiles();

if (je.length > 0) {

he.video\_ids = je;

he.has\_attachment = true;

}

var ke = ee.\_fileUploader.getAudioFiles();

if (ke.length > 0) {

he.audio\_ids = ke;

he.has\_attachment = true;

}

var le = ee.\_fileUploader.getFiles();

if (le.length > 0) {

he.file\_ids = le;

he.has\_attachment = true;

}

yc.sendMessage(he);

ee.\_fileUploader.removeAttachments();

ee.\_shareLinkUploader.clear();

if (!r.get('chat\_react\_input'))

ee.\_getNode('input').value = '';

ee.\_typingDetector && ee.\_typingDetector.resetState();

ee.\_messagesView && ee.\_messagesView.setShouldScrollToBottom(true);

});

}

function rd(ee, fe) {

if (fe === 0)

return;

zd(ee, nb.POST\_PUBLISHED, {

count: fe

});

ee.\_waterfallID = kc.generate();

}

function sd(ee, fe, ge, he) {

dd.getThreadMeta(ee.\_threadID, function(ie) {

var je = yc.constructUserGeneratedMessageObject(fe, fb.CHAT\_WEB, ee.\_threadID);

if (ge.length > 0) {

je.has\_attachment = true;

je.raw\_attachments = ge;

}

var ke = ee.\_shareLinkUploader.getAttachData();

if (ke) {

je.has\_attachment = true;

je.content\_attachment = ke;

}

if (dd.isNewEmptyLocalThread(ee.\_threadID)) {

var le = ee.\_sheetController.getAddFriendsParticipants();

if (le === null || le.length === 0)

return;

le = le.filter(function(ne) {

return ne != kb.getID();

});

if (le.length === 0) {

je.thread\_id = bb.getThreadIDFromUserID(ja.getID());

} else if (le.length === 1) {

je.thread\_id = bb.getThreadIDFromParticipantID(le[0]);

} else

cd.addParticipants(ee.\_threadID, le);

}

if (dd.isEmptyLocalThread(ee.\_threadID)) {

je.message\_id = bb.tokenize(ee.\_threadID).value;

var me = dd.getThreadMetaNow(ie.thread\_id);

je.specific\_to\_list = Array.from(me.participants);

}

if (typeof je != 'undefined')

je.signatureID = ee.\_signatureID;

je.ui\_push\_phase = ha.UIPushPhase;

he(je);

if (ee.\_threadID !== je.thread\_id&&!ee.\_photoUploader.isUploading()) {

ae.inform('closed-tab', ee.\_threadID);

u.openThread(je.thread\_id, 'ChatTabView');

}

});

}

function td(ee) {

ee.\_blocked = true;

ee.\_sheetController.openUploadWarningTabSheet();

}

function ud(ee) {

return (ee.\_fileUploader.isUploading() || ee.\_photoUploader.isUploading());

}

function vd(ee) {

return ee.\_fileUploader.isUploading();

}

function wd(ee) {

var fe = ee.\_getInputValue();

if (!ed.test(fe))

return true;

if (ee.\_fileUploader.getAttachments().length > 0 || ee.\_fileUploader.getImageFiles().length > 0 || ee.\_fileUploader.getVideoFiles().length > 0 || ee.\_fileUploader.getAudioFiles().length > 0 || ee.\_fileUploader.getFiles().length > 0)

return true;

if (ee.\_shareLinkUploader.getAttachData())

return true;

return false;

}

function xd(ee) {

if (ee.\_blocked) {

if (vd(ee))

return;

ee.\_blocked = false;

qd(ee);

ee.\_sheetController.closeUploadWarningTabSheet();

}

}

function yd(ee) {

ee.\_nubController.flyoutContentChanged();

ee.\_attachmentsDiv.scrollTop = ee.\_attachmentsDiv.scrollHeight;

}

function zd(ee, fe, ge) {

nb.sendSignal(nc({

qn: ee.\_waterfallID,

step: fe,

uploader: nb.APP\_CHAT

}, ge || {}));

}

function ae(ee, fe, ge, he, ie) {

if (fe)

zc.ensureThreadIsFetched(fe);

this.\_threadID = ee;

this.\_signatureID = ge;

this.\_eventListeners = [];

this.\_isDragDropActive = false;

this.\_tabTemplate = pd(ee);

this.\_tabElem = this.\_tabTemplate.getRoot();

if (ie) {

this.insertBefore(ie);

} else

this.appendTo(he);

this.\_titlebar = this.\_getNode('nubFlyoutTitlebar');

this.\_iconsContainerNode = this.\_getNode('iconsContainer');

this.\_inputContainerNode = this.\_getNode('inputContainer');

this.\_waterfallID = kc.generate();

this.\_subscriptionsHandler = new ac();

this.\_initializeAutoSendPhotoUploader();

this.\_stickersFlyoutShown = false;

var je = qa.scry(this.\_tabElem, ".\_1ia");

if (je && je.length > 0) {

this.\_dropTarget = new ra(je[0]);

this.\_dropTarget.setOnFilesDropCallback(this.\_filesDropped.bind(this));

this.\_dropTarget.setOnURLDropCallback(this.\_linkDropped.bind(this));

this.\_dropTarget.enable();

}

this.\_attachmentsDiv = this.\_getNode('attachmentShelf');

this.\_sheetController = new da(this.\_threadID, this.\_getNode('sheet'), this.\_tabElem);

if (r.get('chat\_react\_option\_menu')) {

this.\_initializeReactMenu(this.\_tabTemplate.getNode('attachmentShelf'));

} else {

this.\_fileUploader = new cb(this.\_tabTemplate.getNode('attachmentShelf'), this.\_tabTemplate.getNode('attachButtonForm'), this.\_tabTemplate.getNode('fileInput'), this.\_tabTemplate.getNode('attachID'));

this.\_initializeUploader(this.\_fileUploader);

}

this.\_initializeArchiveWarning();

this.\_initializeAddToThreadButton();

this.\_initializeVideoCallButton();

this.\_initializeCloseButton();

this.\_initializeTitle();

this.\_sheetController.onClickNewMessagesSheet(function() {

this.\_messagesView && this.\_messagesView.scrollToBottom();

this.focus();

ae.inform('read', this.\_threadID);

}.bind(this));

this.\_sheetController.onClickThreadIsMutedSheet(function() {

cd.unmute(this.\_threadID);

this.focus();

}.bind(this));

this.\_nubController = new lb().init(this.\_tabElem);

this.\_sheetController.onResize(this.\_nubController.flyoutContentChanged.bind(this.\_nubController));

this.\_sheetController.onTokensChanged(function(pe, qe) {

dd.getThreadMeta(this.\_threadID, function(re) {

this.\_updateControls(re, qe.token\_count);

}.bind(this));

}.bind(this));

if (gd === null)

gd=!r.get('seen\_autosend\_photo\_nux');

this.\_messagesView = null;

if (r.get('chat\_react\_option\_menu')) {

lc.getThreadURI(ee, function(pe) {

return this.\_setTitleHref(pe);

}.bind(this));

} else {

var ke = this.\_getNode('conversationLink');

ka.hide(ke);

bd.renderWebMessengerLink(ee, ke, function() {

ka.show(ke);

this.\_setTitleHref(ke.href);

}.bind(this));

}

if (!bb.getUserIDFromThreadID(this.\_threadID))

this.\_titlebarTooltipAnchor = this.\_getNode('titlebarText');

var le = this.\_getCanonicalUserID();

if (this.\_getCanonicalUserID()) {

eb.get(bb.getParticipantIDFromUserID(le), function(pe) {

if (pe.href)

this.\_setTitleHref(pe.href);

}.bind(this));

} else if (r.get('chat\_react\_option\_menu') && this.\_reactChatTabMenu.getURL()) {

this.\_setTitleHref(this.\_reactChatTabMenu.getURL());

} else if (ke && ke.href)

this.\_setTitleHref(ke.href);

this.\_nubController.subscribe('resize', function() {

this.\_resizeInputContainer();

}.bind(this));

this.\_setNameConversationLink();

this.\_listen('closeButton', 'click', this.\_closeClicked);

this.\_listen('dockButton', 'click', this.\_nubClicked);

this.\_listen('dockButton', 'keydown', this.\_dockKeyDown);

this.\_listen('nubFlyoutTitlebar', 'click', this.\_titleClicked);

this.\_listen('chatConv', 'click', this.\_chatConvClicked);

this.\_listen('inputContainer', 'click', this.\_inputContainerClicked);

this.\_listen('addFriendLink', 'click', this.\_addFriendLinkClicked, true);

this.\_listen('nameConversationLink', 'click', this.\_nameConversationLinkClicked, true);

this.\_listen('clearWindowLink', 'click', this.\_clearHistory, true);

this.\_listen('unsubscribeLink', 'click', this.\_unsubscribeLinkClicked, true);

this.\_listen('reportSpamLink', 'click', this.\_reportSpamClicked, true);

this.\_listen('createGroupLink', 'click', this.\_createGroupClicked, true);

this.\_listen('enableDesktopNotif', 'click', this.\_enableDesktopNotif, true);

this.\_listen('muteThreadLink', 'click', hb.showMuteChangeDialog.bind(null, this.\_threadID), true);

this.\_listen('unmuteThreadLink', 'click', function() {

cd.unmute(this.\_threadID);

ec.hide();

}.bind(this), true);

this.\_listen('switchToWorkUserLink', 'click', this.\_switchToWorkUserClicked, true);

this.\_listen('sheet', 'keydown', function(event) {

if (!event.getModifiers().any && event.keyCode === ya.TAB) {

this.\_focusInput();

event.kill();

}

}.bind(this));

this.\_stickersFlyoutToggler = this.\_getNode('emoticons');

if (ub.WebStickerTrigger)

this.\_stickerTriggerController = new aa(this.\_getNode('stickersTriggerFlyoutContainer'), this.\_getNode('emoticonIndicator'), this.\_stickersFlyoutToggler, this.\_stickerSelected.bind(this), this.\_searchStickers.bind(this));

this.\_initializeInput();

if (xb.shouldShowStickerSearchNUXIndicator()) {

this.\_addStickerSearchNUXIndicator();

this.\_listen('emoticons', 'click', function() {

if (xb.shouldShowStickerSearchNUXIndicator()) {

tb.selectPack(vb.SEARCH\_PACK\_ID);

xb.clearShowStickerSearchNUXIndicator();

}

g.inform('stickerSearchNUXIndicator/hide');

}, true);

g.subscribeOnce('stickerSearchNUXIndicator/hide', function() {

this.\_clearStickerSearchNUXIndicator();

this.\_clearStickerSearchNUXTooltip();

}.bind(this));

ae.subscribe('lower-activated', function(pe, qe) {

if (qe === this.\_threadID && xb.shouldShowStickerSearchNUXIndicator()) {

this.\_clearStickerSearchNUXTooltip();

this.\_addStickerSearchNUXIndicator();

}

}.bind(this));

}

if (r.get('chat\_react\_sticker\_button')) {

this.\_stickerButton = qb.render(qb.createElement(z, {

onFlyoutShown: this.\_onFlyoutShown.bind(this),

onFlyoutHidden: this.\_onFlyoutHidden.bind(this),

onStickerSelect: this.\_stickerSelected.bind(this),

onEmoticonSelect: this.\_emoticonSelected.bind(this)

}), this.\_getNode('stickerButtonContainer'));

m.bumpEntityKey('chat.web', 'sticker\_button.mounting\_attempted');

} else {

var me = this.\_getNode('stickers');

if (me)

this.\_stickersFlyout = qb.render(qb.createElement(wb, {

onStickerSelect: function(pe) {

this.\_stickerSelected(pe);

}.bind(this),

onShown: this.\_onFlyoutShown.bind(this),

onHidden: this.\_onFlyoutHidden.bind(this),

onEmoticonSelect: this.\_emoticonSelected.bind(this),

onEscKeyDown: this.\_hideStickersFlyoutIfShown.bind(this)

}), me);

this.\_eventListeners.push(xb.addListener(xb.PACK\_SELECTED, function(pe) {

if (this.\_stickersFlyout && this.\_stickersFlyout.isMounted())

this.\_stickersFlyout.setProps({

packID: pe

});

}.bind(this)));

}

if (wc) {

this.\_orionTriggerContainer = this.\_getNode('orionTriggerContainer');

this.\_orionTrigger = qb.render(qb.createElement(wc, {

onTrigger: this.\_orionTriggered.bind(this)

}), this.\_orionTriggerContainer);

}

var ne = this.\_getNode('contentRoll');

if (ne)

qb.render(qb.createElement(ia, {

onSelection: function(pe) {

this.\_shareLinkUploader && this.\_shareLinkUploader.loadShare(pe.url, {

content\_roll\_entry: pe

});

this.focus();

}.bind(this),

userID: ja.getID()

}), ne);

this.\_privacyLink = this.\_getNode('privacyLink');

if (this.\_privacyLink) {

this.\_privacyActionController = new w(le, this.\_updatePrivacyLink.bind(this));

this.\_eventListeners.push(ua.listen(this.\_privacyLink, 'click', this.\_privacyActionController.togglePrivacy.bind(this.\_privacyActionController)));

}

dd.getThreadMeta(this.\_threadID, function(pe) {

this.\_setUpMutingSettings(pe);

}.bind(this));

x.removeEmptyHrefs(this.\_tabElem);

fd[ee] = this;

this.updateAvailableStatus();

this.updateTab();

this.\_setCloseTooltip(false);

var oe = {

threadID: ee,

userID: le,

signatureID: this.\_signatureID

};

new k('/ajax/chat/opentab\_tracking.php', oe).send();

rb.onBeforeUnload(function() {

if (this.\_hasUnfinishedPost())

return "Du hast deine Nachricht noch nicht versendet. M\u00f6chtest du abbrechen, ohne sie fertig zu stellen?";

if (yc.getNumberLocalMessages(this.\_threadID))

return "Deine Nachricht wird noch immer versendet. Bist du sicher, dass du das abbrechen m\u00f6chtest?";

return null;

}.bind(this), false);

rb.onUnload(function() {

be(this);

}.bind(this));

}

function be(ee) {

if (ee.\_photoUploader.isUploading())

zd(ee, nb.CANCEL\_DURING\_UPLOAD);

}

function ce() {

for (var ee in fd) {

fd[ee].updateAvailableStatus();

fd[ee].updateMultichatTooltip();

}

}

g.subscribe(['buddylist/availability-changed'], ce);

ob.subscribe(['privacy-changed', 'privacy-availability-changed'], ce);

p.subscribe(p.ON\_CHANGED, function() {

for (var ee in fd)

dd.getThreadMeta(ee, function(fe) {

fd[ee].\_updateUnreadCount(fe);

});

});

jb.addRetroactiveListener('state-changed', function(ee) {

for (var fe in ee) {

var ge = ee[fe] && ee[fe].length, he = fd[fe];

he && he.showTypingIndicator(ge);

}

});

ad.subscribe('threads-updated', function(ee, fe) {

for (var ge in fd) {

var he = fd[ge];

fe[ge] && he.updateTab();

if (he.\_newThreadIDFromPhotoUpload&&!he.\_photoUploader.isUploading() && fe[he.\_newThreadIDFromPhotoUpload]) {

ae.inform('closed-tab', he.\_threadID);

u.openThread(he.\_newThreadIDFromPhotoUpload, 'ChatTabView');

he.\_newThreadIDFromPhotoUpload = null;

}

}

});

ad.subscribe('threads-deleted', function(ee, fe) {

for (var ge in fd)

fe[ge] && ae.inform('thread-deleted', ge);

});

nc(ae, h, {

get: function(ee) {

return fd[ee];

}

});

nc(ae.prototype, {

getThreadID: function() {

return this.\_threadID;

},

showAddFriend: function() {

setTimeout((function() {

this.\_sheetController.openAddFriendsSheet();

}).bind(this), 0);

},

showNameConversation: function(ee) {

setTimeout((function() {

this.\_sheetController.openNameConversationSheet(ee);

}).bind(this), 0);

},

hideAutomaticNameConversation: function() {

setTimeout((function() {

this.\_sheetController.closeAutomaticNameConversationSheet();

}).bind(this), 0);

},

isVisible: function() {

return ka.shown(this.\_tabElem);

},

setVisibleState: function(ee, fe) {

var ge = ka.shown(this.\_tabElem), he = ka.hasClass(this.\_tabElem, 'opened');

ka.conditionShow(this.\_tabElem, ee);

ka.conditionClass(this.\_tabElem, 'opened', fe);

if (!(ge && he) && ee && fe) {

if (!this.\_messagesView)

this.\_messagesView = new ca(this.\_threadID, this.\_sheetController, this.\_getNode('chatConv'), this.\_getNode('conversation'), this.\_getNode('loadingIndicator'), this.\_getNode('lastMessageIndicator'), this.\_getNode('typingIndicator'), this);

this.\_nubController.flyoutContentChanged();

this.\_messagesView.scrollToBottom();

}

if (ge && he&&!(ee && fe))

this.\_contextualDialogController && this.\_contextualDialogController.tabNotActive();

},

focus: function() {

if (ka.hasClass(this.\_tabElem, 'opened')) {

this.\_focusInput();

} else

this.\_getNode('dockButton').focus();

},

isFocused: function() {

var ee = rc();

return mb.byClass(ee, "\_50mz") === this.\_tabElem;

},

hasEmptyInput: function() {

var ee;

if (r.get('chat\_react\_input')) {

if (!this.\_chatInput ||!this.\_chatInput.isMounted())

return false;

ee = this.\_chatInput.getValue();

} else

ee = this.\_getNode('input').value;

return ed.test(ee);

},

getInputElem: function() {

if (r.get('chat\_react\_input')) {

if (!this.\_chatInput ||!this.\_chatInput.isMounted())

return null;

return this.\_chatInput.getDOMNode();

} else

return this.\_getNode('input');

},

setStickersFlyoutPackID: function(ee) {

if (this.\_stickerButton) {

this.\_setPackIDReact(ee);

} else

this.\_setPackIDLegacy(ee);

},

loadShareFromParams: function(ee, fe) {

this.\_shareLinkUploader.loadShareFromParams(ee, fe);

},

\_linkDropped: function(ee) {

this.\_shareLinkUploader.loadShare(ee);

},

\_filesDropped: function(ee) {

if (this.\_fileUploader)

this.\_fileUploader.addDroppedFiles(ee);

},

showStickerSearchNUXTooltip: function() {

if (hd)

return;

hd = true;

this.\_clearStickerSearchNUXIndicator();

this.\_stickerSearchNUXTooltip = qb.render(qb.createElement(yb, {

showIndicatorFunc: this.\_addStickerSearchNUXIndicator.bind(this)

}), this.\_getNode('tooltipNUXContainer'));

},

\_setPackIDLegacy: function(ee) {

var fe = this.\_stickersFlyout;

if (!this.\_stickersFlyoutToggler ||!fe ||!fe.isMounted())

return;

tb.selectPack(ee);

this.\_toggleStickersFlyoutShownLegacy(true);

},

\_setPackIDReact: function(ee) {

if (!this.\_stickerButton.isMounted())

return;

tb.selectPack(ee);

this.\_toggleStickersFlyoutShownReact(true);

},

\_toggleStickersFlyoutShownReact: function(ee) {

this.\_stickerButton.showFlyout();

this.\_stickersFlyoutShown = ee;

},

\_toggleStickersFlyoutShownLegacy: function(ee) {

if (ee) {

ec.show(this.\_stickersFlyoutToggler);

} else

ec.hide(this.\_stickersFlyoutToggler);

this.\_stickersFlyoutShown = ee;

},

\_hideStickersFlyoutIfShown: function() {

if (this.\_stickersFlyoutToggler && this.\_stickersFlyout && this.\_stickersFlyoutShown)

this.\_toggleStickersFlyoutShownLegacy(false);

},

showStickerReplyNUX: function() {

if (hd)

return;

var ee = this.\_getNode('stickerButtonContainer') || this.\_getNode('emoticons');

new j(new mc().getURI()).setRelativeTo(ee).setData({

thread\_id: this.\_threadID

}).setHandler(function(fe) {

xb.clearShowStickerReplyNUX();

}).send();

},

insertBefore: function(ee) {

oa.insertBefore(ee.\_tabElem, this.\_tabElem);

},

appendTo: function(ee) {

oa.appendContent(ee, this.\_tabElem);

},

nextTabIs: function(ee) {

var fe = ee && ee.\_tabElem;

return this.\_tabElem.nextSibling == fe;

},

getScrollTop: function() {

return oa.find(this.\_tabElem, '.fbNubFlyoutBody.scrollable').scrollTop;

},

setScrollTop: function(ee) {

oa.find(this.\_tabElem, '.fbNubFlyoutBody.scrollable').scrollTop = ee;

},

\_unmountComponentAtNodeSafe: function(ee) {

var fe = this.\_getNode(ee);

fe && qb.unmountComponentAtNode(fe);

},

destroy: function() {

this.\_selectionPosition && this.\_selectionPosition.disable();

while (this.\_eventListeners.length)

this.\_eventListeners.pop().remove();

if (this.\_dropTarget)

this.\_dropTarget.disable();

this.\_messagesView && this.\_messagesView.destroy();

this.\_sheetController && this.\_sheetController.destroy();

if (r.get('chat\_react\_option\_menu')) {

qb.unmountComponentAtNode(this.\_getNode('menuContainer'));

} else

this.\_fileUploader && this.\_fileUploader.destroy();

if (r.get('chat\_react\_header')) {

qb.unmountComponentAtNode(this.\_getNode('addToThreadContainer'));

this.\_unmountComponentAtNodeSafe('archiveContainer');

this.\_unmountComponentAtNodeSafe('videoCallButtonContainer');

qb.unmountComponentAtNode(this.\_getNode('closeButtonContainer'));

qb.unmountComponentAtNode(this.\_getNode('titleContainer'));

}

qb.unmountComponentAtNode(this.\_getNode('photoUploaderContainer'));

if (r.get('chat\_react\_sticker\_button')) {

qb.unmountComponentAtNode(this.\_getNode('stickerButtonContainer'));

} else

qb.unmountComponentAtNode(this.\_getNode('stickers'));

if (ub.WebStickerTrigger)

this.\_stickerTriggerController.destroy();

this.\_subscriptionsHandler && this.\_subscriptionsHandler.release();

this.\_contextualDialogController && this.\_contextualDialogController.destroy();

this.\_privacyActionController && this.\_privacyActionController.destroy();

if (r.get('chat\_react\_input')) {

qb.unmountComponentAtNode(this.\_getNode('inputContainer'));

} else

xa.reset(this.\_getNode('input'));

oa.remove(this.\_tabElem);

delete fd[this.\_threadID];

na.unregisterNubController(this.\_nubController);

},

updateAvailableStatus: function() {

dd.getThreadMeta(this.\_threadID, function(ee) {

var fe = l.OFFLINE, ge = false, he = this.\_getCanonicalUserID();

if (he) {

fe = pb.get(he);

ge = jc.availableForCall(he);

} else {

var ie = ee.participants.map(function(ke) {

return bb.getUserIDFromParticipantID(ke);

});

fe = pb.getGroup(ie);

}

if (!fa.isOnline())

fe = l.OFFLINE;

if (he)

this.\_updateCallLink(fe);

ka.conditionClass(this.\_tabElem, "\_5238", fe === l.ACTIVE);

ka.conditionClass(this.\_tabElem, "\_4hwz", ge);

ka.conditionClass(this.\_tabElem, "\_5239", fe === l.MOBILE);

var je = this.\_getNode('presenceIndicator');

switch (fe) {

case l.ACTIVE:

je.setAttribute('alt', "Online");

break;

case l.MOBILE:

je.setAttribute('alt', "Handy");

break;

default:

je.removeAttribute('alt');

break;

}

}.bind(this));

},

updateTab: function() {

this.\_setNameConversationLink();

dd.getThreadMeta(this.\_threadID, function(ee) {

var fe = [this.\_getNode('name')];

if (!r.get('chat\_react\_header')) {

fe.push(this.\_getNode('titlebarText'));

} else if (this.\_title.isMounted())

this.\_title.setProps({

thread: ee

}, function() {

this.\_getNode('chatWrapper').setAttribute('aria-label', this.\_title.getTitle());

}.bind(this));

bd.renderAndSeparatedParticipantList(this.\_threadID, fe, {

names\_renderer: gb.renderShortNames,

check\_length: true

});

this.\_updateControls(ee);

this.\_updateMutingSettings(ee);

this.\_updateUnreadCount(ee);

this.updateMultichatTooltip();

this.\_updateArchiveWarning(ee);

this.\_updateEmployeeAwayWarning(ee);

this.\_updateNewThread(ee);

this.\_updateNameConversationSheet(ee);

this.\_updateReadOnlySheet(ee);

this.\_updateSwitchToWork(ee);

}.bind(this));

},

\_setNameConversationLink: function() {

var ee = this.\_getNode('nameConversationLink');

if (ee)

dd.isEmptyLocalThread(this.\_threadID) ? ka.hide(ee) : ka.show(ee);

},

\_updateNameConversationSheet: function(ee) {

if (dd.canReply(ee.thread\_id)&&!ee.name&&!ee.is\_canonical&&!ee.name\_conversation\_sheet\_dismissed&&!ee.has\_email\_participant&&!ee.read\_only&&!dd.isEmptyLocalThread(ee.thread\_id)) {

this.showNameConversation(true);

} else

this.hideAutomaticNameConversation();

},

\_updateReadOnlySheet: function(ee) {

var fe = dd.canReply(ee.thread\_id), ge = dd.isEmptyLocalThread(ee.thread\_id);

if (!fe&&!ge)

this.\_sheetController.openReadOnlySheet();

},

updateSignatureID: function(ee) {

this.\_signatureID = ee;

},

\_showPhotoNUXIfNecessary: function() {

if (gd) {

gd = false;

new j('/ajax/chat/photo\_nux.php').setRelativeTo(this.\_getNode('photoUploaderContainer')).setData({

threadID: this.\_threadID

}).send();

return true;

}

},

\_setUpMutingSettings: function(ee) {

var fe = hb.isThreadMuted(ee);

if (fe)

this.\_sheetController.openThreadIsMutedTabSheet();

this.\_updateActionMenu(fe);

},

\_updateMutingSettings: function(ee) {

var fe = hb.isThreadMuted(ee), ge;

if (r.get('chat\_react\_option\_menu')) {

ge = this.\_reactChatTabMenu.props.isMuted;

} else

ge = ka.shown(this.\_getNode('unmuteThreadLink').parentNode);

if (fe&&!ge) {

this.\_sheetController.openThreadIsMutedTabSheet();

} else if (!fe && ge)

this.\_sheetController.closeThreadIsMutedTabSheet();

this.\_updateActionMenu(fe);

},

\_updateActionMenu: function(ee) {

if (r.get('chat\_react\_option\_menu')) {

this.\_reactChatTabMenu.isMounted() && this.\_reactChatTabMenu.setProps({

isMuted: ee

});

} else {

ka.conditionShow(this.\_getNode('muteThreadLink').parentNode, !ee);

ka.conditionShow(this.\_getNode('unmuteThreadLink').parentNode, ee);

}

},

\_updateArchiveWarning: function(ee) {

var fe = false;

eb.get(kb.getID(), function(ge) {

fe = ge.employee;

if (fe)

eb.getMulti(ee.participants, this.\_showArchiveWarningIfAllParticipantsAreEmployees.bind(this));

}.bind(this));

},

\_updateEmployeeAwayWarning: function(ee) {

t.updateEmployeeAwayWarning(ee, function() {

this.\_sheetController.openEmployeeAwaySheet();

}.bind(this), function() {});

},

\_updateSwitchToWork: function(ee) {

var fe = false;

eb.get(kb.getID(), function(ge) {

fe = ge.employee;

if (fe)

eb.getMulti(ee.participants, this.\_showSwitchToWorkIfAllParticipantsAreEmployees.bind(this));

}.bind(this));

},

\_updateControls: function(ee, fe) {

var ge = false;

if (ee && ee.thread\_id) {

var he = bb.tokenize(ee.thread\_id);

if (he.type === 'user') {

var ie = he.value;

ge=!zc.isUser(ie);

}

}

var je = dd.canReply(ee.thread\_id) && (!sc(ee.participants)||!!fe);

ka.conditionShow(this.\_inputContainerNode, je);

ka.conditionShow(this.\_iconsContainerNode, je);

if (!r.get('chat\_react\_option\_menu')) {

ka.conditionShow(this.\_getNode('dropdown'), je);

} else if (this.\_reactChatTabMenu.isMounted()) {

this.\_reactChatTabMenu.setProps({

isEmptyChat: dd.isNewEmptyLocalThread(ee.thread\_id),

show: je

});

if (ge)

this.\_reactChatTabMenu.setProps({

showAddFriend: null

});

}

var ke = je&&!ge;

if (r.get('chat\_react\_header')) {

this.\_addToThreadButton.isMounted() && this.\_addToThreadButton.setProps({

shown: ke

});

} else

ka.conditionShow(this.\_getNode('addToThreadLink'), ke);

this.\_resizeInputContainer();

this.\_updateOrionTrigger(je);

},

\_updateOrionTrigger: function(ee) {

if (this.\_orionTrigger && ee) {

var fe = this.getParticipantIDForOrionNUX();

ka.conditionShow(this.\_orionTriggerContainer, fe);

if (fe) {

eb.get(fe, function(ge) {

ge = wa.set(ge, {

userId: bb.getUserIDFromParticipantID(fe)

});

this.\_orionTrigger.setProps({

receiver: ge,

disabled: !ge.orion\_eligible

});

if (ge.orion\_eligible && this.isOrionNUXEligible())

this.showOrionNUXTooltip();

}.bind(this));

} else

this.\_orionTrigger.setProps({

receiver: {}

});

}

},

getParticipantIDForOrionNUX: function() {

var ee = this.\_sheetController.getAddFriendsTabSheet(), fe = ee.getParticipants(), ge = this.\_getCanonicalUserID(), he = null;

if (ge) {

he = bb.getParticipantIDFromUserID(ge);

} else if (fe && fe.length === 1)

he = fe[0];

return he;

},

showOrionNUXTooltip: function() {

if (this.\_orionTrigger && ab.ShowOrionNUXTooltip)

this.\_orionTrigger.setShouldShowNUXTooltip();

},

isOrionNUXEligible: function() {

var ee = this.getParticipantIDForOrionNUX();

return ee && this.isVisible() && ka.hasClass(this.\_tabElem, 'opened');

},

\_resizeInputContainer: function() {

var ee = this.\_tabElem.clientWidth, fe = 2, ge = ee - (this.\_inputContainerNode.clientWidth + fe), he = za.isRTL() ? 'left': 'right';

zb.set(this.\_iconsContainerNode, he, ge + 'px');

if (!r.get('chat\_react\_input'))

dc.getInstance(this.\_getNode('input')).update();

},

\_updateNewThread: function(ee) {

if (dd.isNewEmptyLocalThread(ee.thread\_id))

this.showAddFriend();

},

\_isEmployeesOnly: function(ee) {

for (var fe in ee)

if (!ee[fe].employee)

return false;

return true;

},

\_showArchiveWarningIfAllParticipantsAreEmployees: function(ee) {

var fe = this.\_isEmployeesOnly(ee), ge = r.get('chat\_react\_header'), he = ge ? this.\_getNode('archiveContainer'): this.\_getNode('titanArchiveWarning');

if (he) {

if (this.\_titlebarTooltipAnchor)

ka.conditionClass(this.\_titlebarTooltipAnchor, 'narrowTitleBar', fe);

if (ge) {

this.\_archiveWarning.isMounted() && this.\_archiveWarning.setProps({

shown: fe

});

} else

ka.conditionShow(he, fe);

}

},

\_showSwitchToWorkIfAllParticipantsAreEmployees: function(ee) {

var fe = this.\_isEmployeesOnly(ee);

if (!r.get('chat\_react\_option\_menu')) {

var ge = this.\_getNode('switchToWorkUserLink');

if (ge)

ka.conditionShow(ge, fe);

} else if (this.\_reactChatTabMenu.isMounted())

this.\_reactChatTabMenu.setProps({

hasSwitchToWork: fe

});

},

updateMultichatTooltip: function() {

dd.getThreadMeta(this.\_threadID, function(ee) {

if (!ee.is\_canonical) {

var fe;

if (r.get('chat\_react\_header')) {

fe = this.\_title.getDOMNode();

} else

fe = this.\_titlebarTooltipAnchor;

jd(fe, ee.participants);

}

}.bind(this));

},

\_getNode: function(ee) {

return this.\_tabTemplate.getNode(ee);

},

\_getCanonicalUserID: function() {

return bb.getUserIDFromThreadID(this.\_threadID);

},

\_listen: function(ee, event, fe, ge) {

var he = this.\_getNode(ee);

if (he) {

this.\_eventListeners.push(ua.listen(he, event, fe.bind(this)));

} else if (!ge)

throw new Error('Could not find node "' + ee + '"');

},

\_closePreClicked: function(ee) {

this.\_closeMouseDown = true;

},

\_closeClicked: function(ee) {

be(this);

ae.inform('closed-tab', this.\_threadID);

if (ee instanceof bc) {

ee.stopPropagation();

return;

}

ee.kill();

},

\_closeEnter: function(ee) {

if (ee.keyCode === ya.RETURN)

this.\_closeClicked(ee);

},

\_nubClicked: function(ee) {

if (!this.\_isDragDropActive) {

ee.kill();

return ae.inform('nub-activated', this.\_threadID);

}

},

\_dockKeyDown: function(event) {

if (event.keyCode === ya.RETURN || event.keyCode === ya.SPACE) {

ae.inform('nub-activated', this.\_threadID);

event.kill();

} else

this.\_handleHotkeyPressed(event);

},

\_handleHotkeyPressed: function(event) {

if (event.keyCode === ya.ESC) {

this.\_handleEscape();

if (r.get('chat\_react\_input'))

event = new pa(event);

event.kill();

} else if (event.keyCode === ya.TAB)

this.\_handleTabPressed(event);

},

\_handleTabPressed: function(event) {

if (!event.ctrlKey) {

var ee = ae.inform('tab-pressed', {

id: this.\_threadID,

shiftPressed: event.shiftKey

});

if (r.get('chat\_react\_input'))

event = new pa(event);

!ee && event.kill();

return true;

}

},

\_handleEscape: function() {

if (this.\_stickersFlyoutShown) {

this.\_stickerButton ? this.\_toggleStickersFlyoutShownReact(false) : this.\_toggleStickersFlyoutShownLegacy(false);

return;

}

be(this);

if (this.hasEmptyInput()) {

ae.inform('esc-pressed', this.\_threadID);

} else {

var ee = "M\u00f6chtest du das Chat-Fenster wirklich schlie\u00dfen?", fe = "Du hast deine Nachricht noch nicht versendet.", ge = new ma().setTitle(ee).setBody(fe).setButtons([ma.CLOSE, ma.CANCEL]).setHandler(ae.inform.bind(ae, 'esc-pressed', this.\_threadID, null)).setModal(true).show(), he = ge.getButtonElement('yes');

he && va.set(he);

}

},

\_titleClicked: function(event) {

var ee = event.getTarget();

if (!mb.byClass(ee, 'titlebarText')&&!mb.byClass(ee, 'uiSelector')&&!mb.byClass(ee, 'addToThread')&&!mb.byClass(ee, 'optionMenu')&&!mb.byClass(ee, 'videoicon')&&!mb.byClass(ee, 'close')&&!this.\_isDragDropActive) {

ae.inform('lower-activated', this.\_threadID);

event.kill();

}

},

\_callEnter: function(ee) {

if (ee.keyCode === ya.RETURN)

this.\_callClicked(ee);

},

\_callClicked: function(ee) {

if (ee.target)

ee.target.blur();

var fe = this.\_getCanonicalUserID();

if (jc.availableForCall(fe)) {

var ge = 'chat\_tab\_icon';

if (ee.target && ka.hasClass(ee.target, 'video\_call\_tour'))

ge = 'chat\_tab\_icon\_tour';

ae.inform('video-call-clicked', {

threadID: this.\_threadID,

userID: fe,

clickSource: ge

});

}

if (r.get('chat\_react\_header')) {

ee.stopPropagation();

} else

return false;

},

\_addFriendLinkClicked: function() {

this.showAddFriend();

},

\_nameConversationLinkClicked: function() {

this.showNameConversation();

},

\_showParticipantsClicked: function() {

zc.getServerThreadID(this.\_threadID, function(ee) {

i.send(new j(hc('/ajax/browser/dialog/multichat\_members/').addQueryData({

id: ee

})));

});

},

\_clearHistory: function() {

var ee = dd.getThreadMetaNow(this.\_threadID);

if (ee) {

var fe = this.\_getCanonicalUserID();

zc.clearChat(this.\_threadID, fe, ee.timestamp);

}

},

\_unsubscribeLinkClicked: function() {

var ee = [];

ee.push({

name: 'unsubscribe',

label: "Unterhaltung verlassen",

handler: this.\_unsubscribeToThread.bind(this)

});

ee.push(ma.CANCEL);

new ma().setTitle("Unterhaltung verlassen?").setBody("Du wirst keine Nachrichten von dieser Unterhaltung mehr erhalten und Teilnehmer k\u00f6nnen sehen, dass du die Unterhaltung verlassen hast.").setButtons(ee).show();

},

\_switchToWorkUserClicked: function() {

var ee = [];

ee.push({

name: 'move',

label: "Move Conversation",

handler: this.\_switchToWork.bind(this)

});

ee.push(ma.CANCEL);

new ma().setTitle("Move Conversation to Work Account?").setBody("Do you want to move this conversation so that it shows in your work account instead?").setButtons(ee).show();

},

\_getUserParticipants: function(ee) {

var fe = [];

ee.forEach(function(ge) {

var he = bb.getUserIDFromParticipantID(ge);

he && fe.push(he);

});

return fe;

},

\_createGroupClicked: function() {

dd.getThreadMeta(this.\_threadID, function(ee) {

if (!ee.is\_canonical) {

var fe = hc('/ajax/groups/create\_get.php').addQueryData({

ref: 'web\_messenger\_dock',

parent\_id: '0'

}), ge = this.\_getUserParticipants(ee.participants);

fe.addQueryData({

members: ge

});

if (ee.name)

fe.addQueryData({

name: ee.name

});

i.send(new j(fe));

}

}.bind(this));

},

\_enableDesktopNotif: function() {

Notification.requestPermission();

},

\_reportSpamClicked: function() {

var ee = this.\_getCanonicalUserID(), fe = hc('/ajax/chat/report.php').addQueryData({

id: ee

}).addQueryData({

src: 'top\_report\_link'

});

i.send(new j(fe));

},

\_inputPasted: function(event) {

if (ic.chrome()) {

var ee = event.clipboardData || event.originalEvent.clipboardData;

if (!ee ||!ee.items || ee.items.length < 1)

return;

var fe = ee.items[0].type;

if (fe !== 'image/png')

return;

if (ee.items[0].getAsFile) {

var ge = ee.items[0].getAsFile();

this.\_photoUploader.uploadFile(ge);

}

}

},

\_focusInput: function() {

if (r.get('chat\_react\_input')) {

this.\_chatInput && this.\_chatInput.isMounted() && this.\_chatInput.focus();

} else

this.\_getNode('input').focus();

},

\_focusTab: function() {

ka.addClass(this.\_tabElem, 'focusedTab');

this.tryMarkAsRead();

this.\_contextualDialogController && this.\_contextualDialogController.tabFocused();

if (!hd && this.\_showPhotoNUXIfNecessary())

g.subscribe(['ChatNUX/show', 'ChatNUX/hide'], function(event) {

hd = event === 'ChatNUX/show';

});

this.\_closeMouseDown = false;

this.\_setCloseTooltip(true);

ae.inform('focus-tab', this.\_threadID);

},

\_blurTab: function() {

ka.removeClass(this.\_tabElem, 'focusedTab');

!this.\_closeMouseDown && this.\_setCloseTooltip(false);

},

\_setCloseTooltip: function(ee) {

if (r.get('chat\_react\_header')) {

this.\_closeButton.isMounted() && this.\_closeButton.setProps({

active: ee

});

return;

}

var fe = this.\_getNode('titlebarCloseButton'), ge = ee ? "Bet\u00e4tige zum Schlie\u00dfen die ESC-Taste": "Reiter schlie\u00dfen";

fc.set(fe, ge, 'above');

},

\_inputKeyDown: function(event) {

this.tryMarkAsRead();

if (event.keyCode === ya.RETURN&&!event.shiftKey) {

if (vd(this)) {

td(this);

event.kill && event.kill();

return;

}

qd(this);

if (ub.WebStickerTrigger)

this.\_stickerTriggerController.activateAfterparty();

event.kill && event.kill();

return;

}

if (event.keyCode === ya.DOWN && event.shiftKey && this.hasEmptyInput()) {

ae.inform('lower-activated', this.\_threadID);

event.kill && event.kill();

return;

}

this.\_handleHotkeyPressed(event);

},

\_initializeInput: function() {

if (!r.get('chat\_react\_input')) {

var ee = this.\_inputContainerNode.clientHeight;

dc.getInstance(this.\_getNode('input')).subscribe('resize', function() {

var fe = this.\_inputContainerNode.clientHeight;

if (fe != ee)

this.\_nubController.flyoutContentChanged();

ee = fe;

}.bind(this));

this.\_listen('input', 'focus', this.\_focusTab);

this.\_listen('input', 'blur', this.\_blurTab);

this.\_listen('input', 'paste', this.\_inputPasted);

if (ic.firefox() && la.get() == 'ko\_KR') {

this.\_listen('input', 'keyup', this.\_inputKeyDown);

} else

this.\_listen('input', 'keydown', this.\_inputKeyDown);

this.\_initializeThingsWithInputFn(this.\_tabTemplate.getNode('input'));

} else {

this.\_chatInput = qb.render(qb.createElement(vc, {

initializeThingsWithInputFn: this.\_initializeThingsWithInputFn.bind(this),

inputChanged: function(fe) {

if (ub.WebStickerTrigger)

this.\_stickerTriggerController.updateInput(fe);

}.bind(this),

inputKeyDown: this.\_inputKeyDown.bind(this),

onBlur: this.\_blurTab.bind(this),

onEscape: this.\_handleEscape.bind(this),

onFocus: this.\_focusTab.bind(this),

onTab: this.\_handleTabPressed.bind(this),

openFlyoutFn: this.\_toggleStickersFlyoutShownLegacy.bind(this, true),

stickersFlyoutToggler: this.\_stickersFlyoutToggler,

searchTriggeredWord: function() {

this.\_toggleStickersFlyoutShownLegacy(true);

this.\_searchStickers(this.\_stickerTriggerController.getTriggeredWord());

}.bind(this),

resizeFn: this.\_nubController.flyoutContentChanged.bind(this.\_nubController),

uploadPhotoFn: this.\_photoUploader.uploadFile

}), this.\_getNode('chatInputContainer'));

if (ub.WebStickerTrigger)

this.\_stickerTriggerController.setInput(this.\_chatInput);

}

},

\_initializeThingsWithInputFn: function(ee, fe) {

this.\_shareLinkUploader = new y(this.\_tabTemplate.getNode('attachmentShelf'), ee, this.focus.bind(this), fe);

this.\_initializeUploader(this.\_shareLinkUploader);

dd.getThreadMeta(this.\_threadID, function(ge) {

zc.getServerThreadID(this.\_threadID, function(he) {

this.\_typingDetector = new gc(this.\_getCanonicalUserID(), ee, 'mercury-chat', null, he, fe);

}.bind(this));

}.bind(this));

},

\_getInputValue: function() {

if (r.get('chat\_react\_input')) {

if (!this.\_chatInput ||!this.\_chatInput.isMounted())

return '';

return this.\_chatInput.getValue() || '';

} else

return this.\_tabTemplate.getNode('input').value || '';

},

\_hasUnfinishedPost: function() {

return wd(this) || ud(this);

},

tryMarkAsRead: function() {

if (!this.\_messagesView || this.\_messagesView.isScrolledToBottom()) {

ae.inform('read', this.\_threadID);

return true;

}

return false;

},

\_chatConvClicked: function(ee) {

this.tryMarkAsRead();

if (mb.byTag(ee.getTarget(), 'a') || oa.getSelection())

return;

this.focus();

},

\_inputContainerClicked: function(ee) {

this.tryMarkAsRead();

this.focus();

},

showTypingIndicator: function(ee) {

if (r.get('chat\_closed\_tab\_typing\_indicator\_animated')) {

var fe = oa.find(this.\_tabElem, ".\_54m9");

if (ee) {

this.\_animatedTypingIndicator = qb.render(qb.createElement(ib, null), fe);

this.\_updateAnimatedIndicatorColor();

ka.addClass(this.\_tabElem, "\_54m8");

ka.show(fe);

} else {

ka.removeClass(this.\_tabElem, "\_54m8");

ka.hide(fe);

qb.unmountComponentAtNode(fe);

this.\_animatedTypingIndicator = null;

}

} else

ka.conditionClass(this.\_tabElem, 'typing', ee);

},

\_updateAnimatedIndicatorColor: function() {

if (this.\_animatedTypingIndicator) {

var ee = ka.hasClass(this.\_tabElem, 'highlightTab') ? 'light': 'dark';

this.\_animatedTypingIndicator.setProps({

color: ee

});

}

},

\_updateUnreadCount: function(ee) {

var fe = ee.unread\_count;

if (typeof fe != 'undefined') {

var ge=!!fe && (!p.showsTabUnreadUI || p.showsTabUnreadUI()), he = this.\_getNode('numMessages');

ka.conditionShow(he, ge);

ka.conditionClass(this.\_tabElem, 'highlightTab', ge);

this.\_updateAnimatedIndicatorColor();

nd(this.\_threadID, this.\_tabElem, ge);

oa.setContent(he, fe);

}

},

\_updateCallLink: function(ee) {

var fe = this.\_getCanonicalUserID();

if (fe === ja.getID()) {

this.\_hideVideoCallouts();

if (r.get('chat\_react\_header')) {

this.\_videoCallButton.isMounted() && this.\_videoCallButton.setProps({

shown: false

});

} else

ka.hide(this.\_getNode('videoCallLink'));

return;

}

if (fa.isOnline()) {

var ge = 'fbid:' + fe;

eb.get(ge, function(he) {

var ie = null;

if (jc.availableForCall(fe)) {

ie = uc.\_("Videoanruf mit {firstname} beginnen", {

firstname: he.short\_name

});

} else {

ie = uc.\_("{firstname} ist f\u00fcr Videoanrufe derzeit nicht verf\u00fcgbar", {

firstname: he.short\_name

});

this.\_hideVideoCallouts();

}

this.\_setVideoCallTooltip(ie);

}.bind(this));

} else {

this.\_setVideoCallTooltip("Du musst online sein, um anrufen zu k\u00f6nnen.");

this.\_hideVideoCallouts();

}

},

\_setVideoCallTooltip: function(ee) {

if (r.get('chat\_react\_header')) {

this.\_videoCallButton.isMounted() && this.\_videoCallButton.setProps({

tooltipLabel: ee

});

} else

fc.set(this.\_getNode('videoCallLink'), ee);

},

\_hideVideoCallouts: function() {

this.\_contextualDialogController.hideVideoCallouts();

},

\_updatePrivacyLink: function(ee) {

var fe;

if (ee == w.OFFLINE) {

fe = "Online gehen";

if (r.get('chat\_react\_option\_menu')) {

this.\_reactChatTabMenu.isMounted() && this.\_reactChatTabMenu.setProps({

privacyText: fe

});

} else

oa.setContent(this.\_privacyLink, fe);

return;

}

var ge = this.\_getCanonicalUserID();

if (!ge)

return;

var he = 'fbid:' + ge;

eb.get(he, function(ie) {

if (ee == w.BLOCKED) {

fe = uc.\_("Chat f\u00fcr {name} einschalten", {

name: ie.short\_name

});

} else

fe = uc.\_("Chat f\u00fcr {name} ausschalten", {

name: ie.short\_name

});

if (r.get('chat\_react\_option\_menu')) {

this.\_reactChatTabMenu.isMounted() && this.\_reactChatTabMenu.setProps({

privacyText: fe

});

} else

oa.setContent(this.\_privacyLink, fe);

}.bind(this));

},

\_unsubscribeToThread: function() {

if (!dd.isEmptyLocalThread(this.\_threadID)) {

var ee = yc.constructLogMessageObject(fb.CHAT\_WEB, this.\_threadID, db.UNSUBSCRIBE, {

removed\_participants: [kb.getID()]

});

yc.sendMessage(ee);

}

ae.inform('unsubscribed', this.\_threadID);

return true;

},

\_switchToWork: function() {

if (!dd.isEmptyLocalThread(this.\_threadID)) {

var ee = yc.constructLogMessageObject(fb.CHAT\_WEB, this.\_threadID, db.SWITCH\_TO\_WORK, {

removed\_participants: [kb.getID()]

});

yc.sendMessage(ee);

}

ae.inform('unsubscribed', this.\_threadID);

return true;

},

\_initializeUploader: function(ee) {

this.\_subscriptionsHandler.addSubscriptions(ee.subscribe(['all-uploads-completed', 'upload-canceled'], function() {

xd(this);

}.bind(this)), ee.subscribe('dom-updated', function() {

yd(this);

}.bind(this)), ee.subscribe('submit', function() {

this.\_focusInput();

}.bind(this)));

},

\_initializeReactMenu: function(ee) {

var fe = this.\_getNode('menuContainer'), ge = bb.isMultichat(this.\_threadID), he = dd.isNewEmptyLocalThread(this.\_threadID), ie = this.\_sheetController.openAddFriendsSheet.bind(this.\_sheetController);

if (fe)

this.\_reactChatTabMenu = qb.render(qb.createElement(ba, {

onFileUploaderMounted: this.setFileUploader.bind(this),

updatePrivacyLinkFunc: this.\_updatePrivacyLink.bind(this),

nameConversationFunc: this.\_nameConversationLinkClicked.bind(this),

showParticipantsFunc: this.\_showParticipantsClicked.bind(this),

leaveConversationFunc: this.\_unsubscribeLinkClicked.bind(this),

clearHistoryFunc: this.\_clearHistory.bind(this),

reportSpamFunc: this.\_reportSpamClicked.bind(this),

createGroupFunc: this.\_createGroupClicked.bind(this),

switchToWorkFunc: this.\_switchToWorkUserClicked.bind(this),

isMultichat: ge,

isEmptyChat: he,

showAddFriend: ie,

threadID: this.\_threadID,

attachmentsShelf: ee,

onUploadFinished: xd.bind(this, this),

onUpdateAttachmentsShelf: yd.bind(this, this),

onSubmit: this.\_focusInput.bind(this),

enableDesktopNotif: function() {

this.\_enableDesktopNotif();

}.bind(this)

}), fe);

},

\_initializeArchiveWarning: function() {

if (r.get('chat\_react\_header'))

this.\_archiveWarning = qb.render(qb.createElement(o, {

isFBAtWork: r.IsWorkUser,

shown: true

}), this.\_getNode('archiveContainer'));

},

\_initializeAddToThreadButton: function() {

if (r.get('chat\_react\_header')) {

this.\_addToThreadButton = qb.render(qb.createElement(n, {

onClick: this.\_addFriendLinkClicked.bind(this),

shown: true

}), this.\_getNode('addToThreadContainer'));

} else

this.\_listen('addToThreadLink', 'click', this.\_addFriendLinkClicked, true);

},

\_initializeVideoCallButton: function() {

if (!bb.getUserIDFromThreadID(this.\_threadID))

return;

var ee = null;

if (r.get('chat\_react\_header')) {

this.\_videoCallButton = qb.render(qb.createElement(ga, {

shown: true,

onClick: this.\_callClicked.bind(this),

onKeyUp: this.\_callEnter.bind(this)

}), this.\_getNode('videoCallButtonContainer'));

ee = this.\_videoCallButton.getDOMNode();

} else {

ee = this.\_getNode('videoCallLink');

this.\_listen('videoCallLink', 'click', this.\_callClicked, true);

this.\_listen('videoCallLink', 'keyup', this.\_callEnter, true);

}

this.\_contextualDialogController = new s(this.\_threadID, ee);

},

\_initializeCloseButton: function() {

if (r.get('chat\_react\_header')) {

this.\_closeButton = qb.render(qb.createElement(q, {

onClick: this.\_closeClicked.bind(this),

onKeyUp: this.\_closeEnter.bind(this),

onMouseDown: this.\_closePreClicked.bind(this)

}), this.\_getNode('closeButtonContainer'));

} else {

this.\_listen('titlebarCloseButton', 'click', this.\_closeClicked);

this.\_listen('titlebarCloseButton', 'keyup', this.\_closeEnter);

this.\_listen('titlebarCloseButton', 'mousedown', this.\_closePreClicked);

}

},

\_initializeTitle: function() {

if (r.get('chat\_react\_header')) {

var ee = dd.getThreadMetaNow(this.\_threadID);

this.\_title = qb.render(qb.createElement(ea, {

isNewThread: dd.isNewEmptyLocalThread(this.\_threadID),

thread: ee,

viewer: ja.getID()

}), this.\_getNode('titleContainer'));

}

},

setFileUploader: function(ee) {

this.\_fileUploader = ee;

},

\_initializeAutoSendPhotoUploader: function() {

this.\_photoUploader = qb.render(qb.createElement(v, {

onSubmit: this.\_handlePUSubmit.bind(this),

onAllUploadsComplete: this.\_handlePUAllUploadsCompleted.bind(this),

onLastUploadFail: this.\_handlePULastUploadFailed.bind(this),

onLastUploadCancel: this.\_handlePULastUploadCanceled.bind(this)

}), this.\_getNode('photoUploaderContainer'));

},

\_handlePUSubmit: function(ee, fe) {

zd(this, nb.UPLOAD\_START, fe);

sd(this, '', [], function(ge) {

if (this.\_threadID !== ge.thread\_id) {

this.\_newThreadIDFromPhotoUpload = ge.thread\_id;

if (this.\_messagesView)

this.\_messagesView.setNewThreadID(ge.thread\_id);

}

ge.content\_attachment = null;

yc.updateNewMessage(ge, fe.upload\_id, fe);

yc.sendMessage(ge);

}.bind(this));

this.\_focusInput();

},

\_handlePULastUploadFailed: function(ee, fe) {

zd(this, nb.CLIENT\_ERROR, fe);

de(fe);

var ge = "Hochladen des Fotos fehlgeschlagen", he = "Bitte versuche es noch einmal. Stelle sicher, dass du ein g\u00fcltiges Foto hochl\u00e4dst.";

new ma().setTitle(ge).setBody(he).setButtons(ma.OK).show();

},

\_handlePULastUploadCanceled: function(ee, fe) {

zd(this, nb.CANCEL\_DURING\_UPLOAD, fe);

de(fe);

},

\_handlePUAllUploadsCompleted: function(ee, fe) {

zd(this, nb.ALL\_UPLOADS\_DONE, fe);

var ge = yc.updateMessageAfterUpload(fe.upload\_id, fe);

yc.clearUploadedMessage(fe.upload\_id);

yc.sendMessage(ge);

var he = fe.image\_ids.length || fe.attachments.length;

rd(this, he);

},

\_stickerSelected: function(ee) {

sd(this, '', [], function(fe) {

fe.has\_attachment = true;

fe.sticker\_id = ee;

yc.sendMessage(fe);

this.focus();

}.bind(this));

this.\_messagesView && this.\_messagesView.scrollToBottom();

},

\_emoticonSelected: function(ee) {

var fe = sa.symbols[ee];

if (!fe)

return;

if (r.get('chat\_react\_input')) {

this.\_chatInput.insertEmoticon(fe);

return;

}

this.\_selectionPosition = this.\_selectionPosition || new sb(this.\_getNode('input'));

var ge = dc.getInstance(this.\_getNode('input')), he = ge.getValue(), ie = ta.insertEmoticon(fe, he, this.\_selectionPosition.getPos());

ge.setValue(ie.result);

this.\_selectionPosition.setPos(ie.start, ie.end);

},

\_clearStickerSearchNUXIndicator: function() {

ka.removeClass(this.\_getNode('emoticonIndicator'), 'nux');

},

\_orionTriggered: function() {

sd(this, '', [], function(ee) {

this.focus();

}.bind(this));

this.\_messagesView && this.\_messagesView.scrollToBottom();

},

\_addStickerSearchNUXIndicator: function() {

ka.addClass(this.\_getNode('emoticonIndicator'), 'nux');

hd = false;

},

\_clearStickerSearchNUXTooltip: function() {

this.\_stickerSearchNUXTooltip && this.\_stickerSearchNUXTooltip.hideNUX();

},

\_onFlyoutShown: function(ee) {

if (r.get('chat\_react\_sticker\_button')) {

var fe = this.\_stickersFlyout.props.packID;

if (fe !== vb.SEARCH\_PACK\_ID)

this.focus();

}

if (ub.WebStickerTrigger) {

this.\_stickerTriggerController.onFlyoutShown(ee);

} else

ee();

this.\_stickersFlyoutShown = true;

},

\_searchStickers: function(ee, fe) {

this.\_stickersFlyout.isMounted() && this.\_stickersFlyout.setProps({

packID: vb.SEARCH\_PACK\_ID,

trigger: ee

}, fe);

},

\_onFlyoutHidden: function() {

this.\_stickersFlyoutShown = false;

this.focus();

},

\_setTitleHref: function(ee) {

if (r.get('chat\_react\_header')) {

this.\_title.isMounted() && this.\_title.setProps({

href: ee

});

} else {

var fe = this.\_getNode('titlebarText');

fe.setAttribute('href', ee);

fe.removeAttribute('rel');

ka.removeClass(fe, 'noLink');

}

}

});

function de(ee) {

var fe = yc.updateMessageAfterUpload(ee.upload\_id, ee);

yc.clearUploadedMessage(ee.upload\_id);

xc["delete"](fe.thread\_id, [fe.message\_id]);

yc.deleteLocalMessage(fe.thread\_id);

}

e.exports = ae;

}, null);

\_\_d("MercuryNotificationRenderer", ["MercuryAssert", "MercuryParticipants", "MercuryViewer", "tx", "MercuryMessages", "MercuryThreads"], function(a, b, c, d, e, f, g, h, i, j) {

var k = b('MercuryMessages').get(), l = b('MercuryThreads').get();

function m(n, o) {

g.isThreadID(n);

l.getThreadMeta(n, function(p) {

k.getThreadMessagesRange(n, 0, 1, function(q) {

var r = q.length && q[q.length - 1];

if (r && r.author != i.getID()) {

h.get(r.author, function(s) {

if (p.name) {

o(j.\_("{senderName} hat {groupName} eine Nachricht geschickt", {

senderName: s.short\_name,

groupName: p.name

}));

} else

o(j.\_("{name} hat dir eine Nachricht geschickt", {

name: s.short\_name

}));

});

} else

o("Neue Nachricht!");

});

});

}

e.exports = {

renderDocumentTitle: m

};

}, null);

\_\_d("ChatTitleBarBlinker", ["ChatActivity", "DocumentTitle", "JSLogger", "MercuryNotificationRenderer", "PresenceState", "MercuryThreadInformer", "MercuryTimestampTracker"], function(a, b, c, d, e, f, g, h, i, j, k) {

var l = b('MercuryThreadInformer').get(), m = b('MercuryTimestampTracker').get(), n = i.create('chat\_title'), o = null, p = 0, q = false;

function r() {

if (o) {

o.stop();

o = null;

return true;

}

return false;

}

function s(x) {

var y = x || m.getLastUserMessageTimestamp();

if (p <= y) {

p = y;

if (r() || q)

k.doSync();

}

}

var t = {

blink: function(x, y) {

if (!o && p < y)

j.renderDocumentTitle(x, function(z) {

if (!o)

o = h.blink(z);

});

},

stopBlinking: function() {

s();

},

blinkingElsewhere: function() {

q = true;

}

};

function u(x) {

var y = k.verifyNumber(x.sb2);

if (!y || y <= p)

return null;

return y;

}

function v(x) {

var y = x && u(x);

if (y) {

p = y;

n.debug('load', p);

r();

q = false;

}

}

function w(x) {

var y = u(x);

if (!y) {

n.debug('store', p);

x.sb2 = p;

q = false;

}

return x;

}

k.registerStateStorer(w);

k.registerStateLoader(v);

l.subscribe('thread-read-changed', function(x, y) {

var z = m.getLastUserMessageTimestamp(), aa = 0;

for (var ba in y)

if (y[ba].mark\_as\_read && y[ba].timestamp >= z && y[ba].timestamp > aa)

aa = y[ba].timestamp;

aa && s(aa);

});

g.subscribe('activity', function() {

s();

});

(function() {

var x = k.getInitial();

if (x)

p = u(x) || 0;

})();

e.exports = t;

}, null);

\_\_d("ChatNewMessageHandler", ["ArbiterMixin", "ChatActivity", "ChatMentionsNotifications", "ChatTabModel", "ChatTabView", "JSLogger", "MercuryAssert", "MercuryBrowserAlerts", "UserActivity", "MercuryConfig", "MercuryThreadlistConstants", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r) {

var s = l.create('chat\_new\_message'), t = {

\_raiseNewMessageTab: function(u, v) {

var w = j.getTab(u), x = false;

if (w) {

x = w.raised;

} else

x = true;

t.inform('chat/new-message-from-server', {

thread\_id: u

});

v.to\_new\_tab=!w;

v.to\_raised\_tab=!!x;

},

\_notify: function(u, v, w) {

if (p.DesktopNotificationsGK)

i.notifyIfMessageToMe(v);

var x = k.get(u);

w.view\_is\_visible = x && x.isVisible();

w.view\_is\_focused = x && x.isFocused();

if (!w.view\_is\_visible)

s.log('message\_to\_hidden');

w.is\_active = h.isActive();

n.messageReceived(v);

},

newMessage: function(u, v, w) {

m.isThreadID(u);

var x = {

thread\_id: u,

message\_id: v.message\_id

};

this.\_raiseNewMessageTab(u, x);

this.\_notify(u, v, x);

s.log('message', x);

}

};

r(t, g);

n.subscribe('before-alert', function(u, event) {

var v = event.threadID, w = k.get(v), x = j.getTab(v), y = q.MESSAGE\_NOTICE\_INACTIVITY\_THRESHOLD;

if (x && x.raised && w && w.isVisible() && w.isFocused() && o.isActive(y)) {

w.tryMarkAsRead();

event.cancelAlert();

}

});

e.exports = t;

}, null);

\_\_d("ChatTabLRUManager", ["ChatTabView", "ChatTabModel", "MercuryThreads"], function(a, b, c, d, e, f, g, h) {

var i = b('MercuryThreads').get(), j = {

getLRUVisibleTab: function(k) {

if (k.hasRoomForRaisedTab())

return undefined;

var l = Object.keys(k.getTabsToShow() || {}), m = 1 \* 60, n = null, o = Infinity;

for (var p = 0; p < l.length; p++) {

var q = l[p], r = i.getThreadMetaNow(q);

if (h.isTabPromoted(q))

continue;

if (!g.get(q).hasEmptyInput() ||!r)

continue;

var s = (h.getServerTime() - r.timestamp) / 1000;

if (!r.timestamp || (r.timestamp && r.timestamp < o && s > m)) {

n = r.thread\_id;

o = r.timestamp;

}

}

return n;

}

};

e.exports = j;

}, null);

\_\_d("ChatTabPolicy", ["ChatBehavior", "JSLogger", "MercuryActionTypeConstants", "MercuryLogMessageType", "MercuryAssert", "MercuryIDs", "MercuryParticipantTypes", "MercurySourceType", "MercuryThreadMode", "MercuryThreadMuter", "MercuryViewer", "MessagingTag", "PresencePrivacy", "ShortProfiles", "MercuryUnseenState"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t) {

var u = b('MercuryUnseenState').get(), v = h.create('tab\_policy');

function w(x, y) {

u.markThreadSeen(x, y);

}

e.exports = {

messageIsAllowed: function(x, y, z) {

var aa = x.thread\_id, ba = y.message\_id;

k.isThreadID(aa);

k.isParticipantID(y.author);

var ca;

if (p.isThreadMuted(x)) {

ca = {

thread\_id: aa,

message\_id: ba,

mute\_settings: x.mute\_settings

};

v.log('message\_thread\_muted', ca);

return;

}

if (x.mode == o.OBJECT\_ORIGINATED) {

ca = {

thread\_id: aa,

message\_id: ba,

mode: x.mode

};

v.log('message\_object\_originated', ca);

return;

}

if (y.source == n.EMAIL || y.source == n.TITAN\_EMAIL\_REPLY) {

ca = {

thread\_id: aa,

message\_id: ba,

source: y.source

};

v.log('message\_source\_not\_allowed', ca);

return;

}

var da = l.getUserIDFromParticipantID(y.author);

if (!da) {

v.log('message\_bad\_author', {

thread\_id: aa,

message\_id: ba,

user: da

});

return;

}

if (y.action\_type != i.USER\_GENERATED\_MESSAGE&&!this.\_messageIsNewlyCreatedGroup(y, x)) {

ca = {

thread\_id: aa,

message\_id: ba,

type: y.action\_type

};

v.log('message\_non\_user\_generated', ca);

return;

}

if (x.is\_canonical\_user&&!g.notifiesUserMessages()) {

v.log('message\_jabber', {

thread\_id: aa,

message\_id: ba

});

w(aa, true);

return;

}

if (x.is\_canonical&&!x.canonical\_fbid) {

v.log('message\_canonical\_contact', {

thread\_id: aa,

message\_id: ba

});

return;

}

if (x.folder != r.INBOX) {

v.log('message\_folder', {

thread\_id: aa,

message\_id: ba,

folder: x.folder

});

return;

}

var ea = l.getUserIDFromParticipantID(q.getID());

t.getMulti([da, ea], function(fa) {

if (!s.allows(da)) {

v.log('message\_privacy', {

thread\_id: aa,

message\_id: ba,

user: da

});

return;

}

var ga = fa[da].employee && fa[ea].employee;

if (!ga && fa[da].type !== m.FRIEND) {

v.log('message\_non\_friend', {

thread\_id: aa,

message\_id: ba,

user: da

});

return;

}

z();

});

},

\_messageIsNewlyCreatedGroup: function(x, y) {

return x.action\_type === i.LOG\_MESSAGE && x.log\_message\_type === j.THREAD\_NAME && y.message\_count === 1;

}

};

}, null);

\_\_d("ChatTabViewSelector", ["Event", "Arbiter", "CSS", "DataStore", "DOM", "MenuDeprecated", "NubController", "ChatTabTemplates", "Toggler", "copyProperties", "MercuryThreadInformer", "MercuryThreads", "MercuryThreadMetadataRenderer"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p) {

var q = b('MercuryThreadInformer').get(), r = b('MercuryThreads').get(), s = b('MercuryThreadMetadataRenderer').get();

function t(u) {

var v = n[':fb:chat:tab:selector'].build(), w = v.getRoot(), x = v.getNode('menu'), y = k.find(x, '.uiMenuInner'), z = {}, aa = new m().init(w);

i.hide(w);

k.insertBefore(u, w);

function ba(ea) {

var fa = 0;

for (var ga in z) {

var ha = z[ga], ia = r.getThreadMetaNow(ga), ja = ha.getNode('unreadCount'), ka = (ia && ia.unread\_count) || 0;

fa += ka;

if (ka > 9)

ka = '+';

i.conditionClass(ja, 'invisible\_elem', !ka);

k.setContent(ja, ka);

}

var la = v.getNode('numMessages');

i.conditionShow(la, fa);

k.setContent(la, fa);

}

this.setTabData = function(ea) {

z = {};

if (ea.length < 1) {

i.hide(w);

return;

}

i.show(w);

k.empty(y);

ea.forEach(function(fa) {

var ga = n[':fb:chat:tab:selector:item'].build();

z[fa.id] = ga;

var ha = ga.getNode('content');

s.renderAndSeparatedParticipantList(fa.id, ha);

k.prependContent(y, ga.getRoot());

j.set(ga.getRoot(), 'threadID', fa.id);

var ia = ga.getNode('closeButton');

g.listen(ia, 'click', function(event) {

t.inform('selector/close-tab', fa.id);

event.kill();

});

});

aa.flyoutContentChanged();

k.setContent(v.getNode('numTabs'), ea.length);

ba();

};

function ca(event, ea) {

if (ea.menu != x)

return;

var fa = j.get(ea.item, 'threadID');

t.inform('selected', fa);

o.hide(w);

}

function da(event, ea) {

l.register(x);

}

l.subscribe('select', ca.bind(this));

o.listen('show', w, function() {

h.inform('layer\_shown', {

type: 'ChatTabSelector'

});

da();

});

o.listen('hide', w, function() {

h.inform('layer\_hidden', {

type: 'ChatTabSelector'

});

});

q.subscribe('threads-updated', ba);

}

p(t, new h());

e.exports = t;

}, null);

\_\_d("ChatTabController", ["Arbiter", "ChatActivity", "ChatConfig", "ChatNewMessageHandler", "ChatTabLRUManager", "ChatTabMessagesView", "ChatTabModel", "ChatTabPolicy", "ChatTabView", "ChatTabViewSelector", "JSLogger", "MercuryConfig", "MercuryIDs", "MercuryParticipants", "MercuryViewer", "StickerState", "Style", "UserAgent\_DEPRECATED", "VideoCallCore", "ChatTabPresence", "ChatTypeaheadBehavior", "MercuryThreadInformer", "MercuryThreadActions", "MercuryThreads", "MercuryUnseenState"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y) {

b('ChatTabPresence');

b('ChatTypeaheadBehavior');

var z = b('MercuryThreadInformer').get(), aa = b('MercuryThreadActions').get(), ba = b('MercuryThreads').get(), ca = b('MercuryUnseenState').get(), da = i.get('tab\_auto\_close\_timeout') || 7200000, ea = q.create('tab\_controller');

function fa(ta) {

aa.markRead(ta);

ga(ta);

}

function ga(ta) {

ca.markThreadSeen(ta);

}

function ha(ta, ua, va) {

var wa = m.get().tabs;

ta += ua ? 1 : - 1;

while (ta >= 0 && ta < wa.length) {

var xa = wa[ta], ya = o.get(xa.id);

if (ya && ya.isVisible() && (!va || xa.raised)) {

ya.focus();

return true;

}

ta += ua ? 1 : - 1;

}

return false;

}

function ia(ta, ua) {

var va = ua && ua.getRightmostHiddenTab();

if (va === ta)

return null;

return va;

}

function ja() {

return Math.floor(Math.random() \* 2147483648).toString(16);

}

function ka(ta, ua, va, wa) {

m.raiseTab(ta, true, va);

var xa = ua && ua.getLeftmostVisibleTab(), ya = ua && ua.getTabsToShow();

if (ya&&!ya[ta] && xa) {

m.promoteThreadInPlaceOfThread(ta, xa, ia(ta, ua), true);

} else

m.setPromotedTab(ta);

g.inform('chat/open-tab', ta);

var za = o.get(ta);

za && za.focus();

}

function la(ta, ua, va) {

var wa = va.forceDrop();

if (!m.getTab(ta))

m.raiseTab(ta, false);

if (wa)

return;

var xa = ua && k.getLRUVisibleTab(ua), ya = ua && ua.getTabsToShow();

if (ya&&!ya[ta] && xa)

m.promoteThreadInPlaceOfThread(ta, xa, ia(ta, ua));

}

function ma(ta, ua) {

var va = m.indexOf(ta);

m.closeTab(ta, ua);

return va;

}

function na(ta, ua) {

j.subscribe('chat/new-message-from-server', function(db, eb) {

la(eb.thread\_id, ta, ua);

});

p.subscribe('selected', function(db, eb) {

ka(eb, ta);

});

g.subscribe('chat/tabs-order-changed', function(db, eb) {

m.changeShowingTabs(eb);

});

g.subscribe('page\_transition', function(db, eb) {

m.closeFragileTabs();

});

g.subscribe('chat/close-all-tabs', function() {

m.closeAllTabs();

});

g.subscribe('chat-visibility/go-offline', function() {

m.closeAllTabs();

});

g.subscribe('chat/close-tab', function(db, eb, fb) {

m.closeTab(eb, fb);

});

o.subscribe('read', function(event, db) {

fa(db);

});

h.subscribe('idle', function(db, eb) {

if (eb > da) {

var fb = m.get().tabs;

fb.forEach(function(gb) {

var hb = gb.id;

ba.getThreadMeta(hb, function(ib) {

if (!ib.unread\_count) {

ea.log('autoclose\_idle\_seen', {

thread\_id: hb,

idleness: eb

});

m.closeTab(hb, 'autoclose\_idle\_seen');

}

});

});

}

});

o.subscribe('nub-activated', function(db, eb) {

ka(eb, ta);

});

o.subscribe('lower-activated', function(db, eb) {

m.lowerTab(eb);

var fb = o.get(eb);

fb && fb.focus();

});

o.subscribe('focus-tab', function(db, eb) {

var fb = m.getTab(eb);

if (fb && fb.raised)

ka(eb, ta, fb.signatureID, true);

});

function va(db, eb) {

y.showOutgoingCallDialog(eb.userID, eb.clickSource);

}

o.subscribe('video-call-clicked', va);

l.subscribe('video-call-clicked', va);

l.subscribe('interaction-with-tab', function(db, eb) {

var fb = m.getTab(eb);

fb && fb.raised && ga(eb);

});

o.subscribe('closed-tab', function(db, eb) {

ea.log('close\_view', {

thread\_id: eb

});

ma(eb, 'close\_view');

return false;

});

o.subscribe('thread-deleted', function(db, eb) {

ea.log('close\_thread\_deleted', {

thread\_id: eb

});

ma(eb, 'close\_thread\_deleted');

return false;

});

o.subscribe('unsubscribed', function(db, eb) {

ea.log('close\_view\_unsubscribed', {

thread\_id: eb

});

ma(eb, 'close\_view\_unsubscribed');

return false;

});

o.subscribe('esc-pressed', function(db, eb) {

ea.log('close\_esc', {

thread\_id: eb

});

var fb = ma(eb, 'close\_esc');

setTimeout(function() {

ha(fb - 1, true, true) || ha(fb, false, true);

}, 0);

});

p.subscribe('selector/close-tab', function(db, eb) {

ea.log('close\_chat\_from\_selector', {

thread\_id: eb

});

ma(eb, 'close\_chat\_from\_selector');

});

z.subscribe('messages-received', function(db, eb) {

for (var fb in eb) {

var gb = eb[fb];

for (var hb = 0; hb < gb.length; hb++) {

var ib = gb[hb];

if (ib.author != u.getID()) {

if (!ib.is\_unread) {

ea.log('message\_already\_read', {

action\_id: ib.action\_id,

thread\_id: ib.thread\_id

});

continue;

}

ba.getThreadMeta(fb, function(jb) {

n.messageIsAllowed(jb, ib, function() {

j.newMessage(fb, ib, ta);

});

});

}

}

}

});

z.subscribe('thread-read-changed', function(db, eb) {

for (var fb in eb)

if (!eb[fb].mark\_as\_read) {

ea.log('autoclose\_marked\_unread', {

thread\_id: fb

});

m.closeTab(fb, 'autoclose\_marked\_unread');

}

});

o.subscribe('tab-pressed', function(db, eb) {

return !ha(m.indexOf(eb.id), eb.shiftPressed);

});

g.subscribe(q.DUMP\_EVENT, function(db, eb) {

eb.chat\_controller = {

auto\_close\_timeout: da

};

});

this.openTab = function(db, eb) {

ka(db, eb, ja());

};

var wa, xa, ya = [], za = [], ab = [], bb;

if (v.shouldShowStickerSearchNUXTooltip()) {

wa = ta.getRightmostRaisedTab();

if (wa) {

bb = o.get(wa);

bb.showStickerSearchNUXTooltip();

} else

g.subscribeOnce('chat/open-tab', function(db, eb) {

bb = o.get(eb);

bb.showStickerSearchNUXTooltip();

});

} else if (r.ShowOrionNUXTooltip) {

ya = ta.getRaisedVisibleTabs();

for (var cb = 0; cb < ya.length; cb++) {

xa = o.get(ya[cb]);

if (xa.isOrionNUXEligible())

za.push(xa);

}

if (za.length) {

ab = za.map(function(db) {

return db.getParticipantIDForOrionNUX();

});

t.getMulti(ab, function(db) {

for (var eb in db)

if (db[eb].orion\_eligible) {

bb = za.filter(function(fb) {

return fb.\_getCanonicalUserID() === s.getUserIDFromParticipantID(db[eb].id);

})[0];

break;

}

if (bb)

bb.showOrionNUXTooltip();

});

}

}

}

if (x.firefox()) {

var oa = function() {

return w.get(document.body, 'overflowX') + ' ' + w.get(document.body, 'overflowY');

}, pa = oa(), qa = function() {

var ta = oa();

if (ta !== pa) {

pa = ta;

g.inform('overflow-applied-to-body');

}

};

if ('MutationObserver' in window) {

var ra = new MutationObserver(qa), sa = {

attributes: true,

attributeFilter: ['class', 'style']

};

ra.observe(document.documentElement, sa);

} else

document.documentElement.addEventListener('DOMAttrModified', function(event) {

if (event.getTarget() === document.documentElement && (event.attrName === 'class' || event.attrName === 'style'))

qa();

}, false);

}

e.exports = na;

}, null);

\_\_d("ChatTabViewCoordinator", ["Arbiter", "BanzaiODS", "Bootloader", "ChatTabModel", "ChatTabView", "ChatTabViewSelector", "CSS", "JSLogger", "Style", "Rect", "VideoCallSupport", "copyProperties"], function(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r) {

var s = n.create('chat\_tab\_coordinator'), t, u;

function v(w, x) {

var y = new l(w), z = {}, aa = {}, ba = false;

function ca() {

var oa = j.get(), pa = {};

oa.tabs.forEach(function(ra) {

pa[ra.id] = 1;

});

s.log('process\_tab\_changed', {

model\_tabs: Object.keys(pa),

view\_tabs: Object.keys(z)

});

for (var qa in z)

if (!pa[qa]) {

s.log('remove\_tab', {

tab\_id: qa

});

z[qa].destroy();

delete (z[qa]);

}

da(oa);

ma(oa);

if (oa.tabNeedsFocus&&!oa.tabNeedsFocus.isFocused())

oa.tabNeedsFocus.focus();

}

function da(oa) {

var pa = null;

oa.tabs.forEach(function(qa) {

var ra = qa.id, sa = false;

if (!z[ra]) {

s.log('add\_tab', {

tab\_id: ra

});

z[ra] = new k(ra, qa.server\_id, qa.signatureID, w, pa);

sa = true;

} else

z[ra].updateSignatureID(qa.signatureID);

if (z[ra].isFocused())

oa.tabNeedsFocus = z[ra];

if (!sa&&!z[ra].nextTabIs(pa)) {

var ta = z[ra].getScrollTop();

if (pa) {

z[ra].insertBefore(pa);

} else

z[ra].appendTo(w);

if (ta)

z[ra].setScrollTop(ta);

}

pa = z[ra];

});

}

function ea() {

var oa = x.getShowingTabsOrder();

if (!oa ||!oa.length)

return;

var pa = z[oa[0].id].\_tabElem, qa = z[oa[oa.length - 1].id].\_tabElem, ra = parseInt(o.get(qa, 'margin-left'), 10) || 0, sa = p.getElementBounds(qa).l - 2 \* ra;

sa = sa > 0 ? sa : 0;

t && t.setBoundingBox(new p(0, p.getElementBounds(pa).r, 0, sa));

}

function fa(oa, pa) {

var qa = false;

pa.forEach(function(ta) {

if (ta.raised) {

if (aa[ta.id]) {

qa = true;

delete aa[ta.id];

}

} else if (!aa[ta.id]) {

qa = true;

aa[ta.id] = true;

}

});

if (qa)

return true;

if (!t)

return true;

var ra = ga() || [];

if (ra.length !== oa.length)

return true;

for (var sa = 0; sa < oa.length; sa++)

if (ra[sa] !== oa[sa].id)

return true;

return false;

}

function ga() {

var oa = t && t.getOrder();

oa && oa.reverse();

return oa;

}

function ha() {

var oa = ga();

if (oa) {

h.bumpEntityKey('chat.web', 'tab.drag\_order\_changed');

g.inform('chat/tabs-order-changed', oa);

}

}

function ia(oa, pa, qa) {

if (z[qa].\_scrollTop != null)

z[qa].setScrollTop(z[qa].\_scrollTop);

}

function ja(oa, pa) {

u = pa;

z[pa].\_isDragDropActive = true;

z[pa].\_scrollTop = z[pa].getScrollTop();

o.set(z[pa].\_tabElem, 'z-index', 1);

ea();

h.bumpEntityKey('chat.web', 'tab.drag\_started');

}

function ka(oa) {

o.set(z[oa].\_tabElem, 'z-index', '');

if (z[oa].\_scrollTop != null)

z[oa].setScrollTop(z[oa].\_scrollTop);

z[oa].\_scrollTop = null;

u = null;

setTimeout(function() {

z[oa].\_isDragDropActive = false;

}, 0);

}

function la(oa) {

i.loadModules(["SortableGroup"], function(pa) {

var qa = x.getShowingTabsOrder(), ra = fa(qa, oa);

if (qa.length <= 1) {

t && t.removeAllSortables();

return;

}

if (ra) {

if (t) {

t.removeAllSortables();

} else

t = new pa().setBeforeGrabCallback(ja).setDragOverCallback(ia).setDropCallback(ka).setOrderChangeHandler(ha).setUseScroller(false).setDragOverHorizontally();

for (var sa = 0; sa < qa.length; sa++) {

var ta = qa[sa];

t.addSortable(ta.id, z[ta.id].\_tabElem, aa[ta.id] ? null : z[ta.id].\_titlebar);

}

}

});

}

function ma(oa) {

var pa = x.getTabsToShow(oa), qa = [], ra = false;

oa.tabs.forEach(function(sa) {

if (!pa[sa.id]) {

z[sa.id].setVisibleState(false, sa.raised);

qa.push(sa);

}

});

oa.tabs.forEach(function(sa) {

if (pa[sa.id]) {

z[sa.id].setVisibleState(true, sa.raised);

ra|=sa.raised;

}

});

y.setTabData(qa);

na(ra);

la(oa.tabs);

}

function na(oa) {

if (!oa && ba) {

g.inform('layer\_hidden', {

type: 'ChatTab'

});

ba = false;

} else if (oa&&!ba) {

g.inform('layer\_shown', {

type: 'ChatTab'

});

ba = true;

}

}

if (q.isVideoCallSupported())

m.addClass(w, 'videoCallEnabled');

x.subscribe('tabs-changed', ca);

ca();

}

r(v.prototype, {

forceDrop: function() {

if (t && u) {

t.forceDrop(u);

u = null;

return true;

}

return false;

}

});

e.exports = v;

}, null);

\_\_d("MercuryStateCheck", ["Arbiter", "ChannelConstants", "MercuryFolders", "MessagingTag", "URI", "copyProperties", "MercuryServerRequests"], function(a, b, c, d, e, f, g, h, i, j, k, l) {

var m = b('MercuryServerRequests').get(), n = l(new g(), {

initialize: function() {

g.subscribe(h.ON\_INVALID\_HISTORY, o);

d(['ChannelConnection'], function(p) {

p.subscribe(p.CONNECTED, function(q, r) {

if (!r.init)

o();

});

});

}

});

function o() {

var p;

if (k.getRequestURI().getPath().search(/messages/)!==-1) {

p = i.getSupportedFolders();

} else

p = [j.INBOX];

m.fetchMissedMessages(p);

}

n.initialize();

e.exports = n;

}, null);

\_\_d("ChatApp", ["CSS", "JSLogger"], function(a, b, c, d, e, f, g, h) {

var i = {

init: function(j, k, l) {

if (this.\_isInitialized) {

h.create('chat\_app').error('repeated\_init');

return;

}

this.\_rootElem = j;

d(['TabsViewport', 'ChatTabController', 'ChatTabViewCoordinator', 'MercuryServerRequests', 'MercuryChannelHandler', 'MercuryStateCheck'], function(m, n, o, p, q, r) {

q = q.get();

q.turnOn();

p.get().handleUpdate(l);

this.tabsViewport = new m(k);

this.tabViewCoordinator = new o(k, this.tabsViewport);

this.tabController = new n(this.tabsViewport, this.tabViewCoordinator);

this.\_isInitialized = this.\_isLoaded = true;

}.bind(this));

},

isInitialized: function() {

return !!this.\_isInitialized;

},

isHidden: function() {

return !this.\_isInitialized || this.\_isHidden;

},

hide: function() {

if (this.isHidden())

return;

g.hide(this.\_rootElem);

this.\_isHidden = true;

},

unhide: function() {

if (this.\_isInitialized) {

if (this.\_isHidden) {

g.show(this.\_rootElem);

this.tabsViewport.checkWidth();

d(['Dock'], function(j) {

j.resizeAllFlyouts();

});

this.\_isHidden = false;

}

} else if (!this.\_isLoaded) {

d(['UIPagelet'], function(j) {

j.loadFromEndpoint('ChatTabsPagelet', 'ChatTabsPagelet');

j.loadFromEndpoint('BuddylistPagelet', 'BuddylistPagelet');

});

this.\_isLoaded = true;

}

}

};

e.exports = i;

}, null);

\_\_d("legacy:CompactTypeaheadRenderer", ["CompactTypeaheadRenderer"], function(a, b, c, d) {

if (!a.TypeaheadRenderers)

a.TypeaheadRenderers = {};

a.TypeaheadRenderers.compact = b('CompactTypeaheadRenderer');

}, 3);