```
npx create-next-app <your_app_name>
```

```
yarn add livekit-server-sdk @livekit/components-react @livekit/components-styles
```

Wir müssen in sunriza26 neue directories anlegen
api/token/route.ts

Create a new file at /app/api/token/route.ts with the following content:

```
import { NextRequest, NextResponse } from 'next/server';
import { AccessToken } from 'livekit-server-sdk';

// Do not cache endpoint result
export const revalidate = 0;

export async function GET(req: NextRequest) {
  const room = req.nextUrl.searchParams.get('room');
  const username = req.nextUrl.searchParams.get('username');
  if (!room) {
    return NextResponse.json({ error: 'Missing "room" query parameter' },
{ status: 400 });
  } else if (!username) {
    return NextResponse.json({ error: 'Missing "username" query parameter' },
{ status: 400 });
  }

  const apiKey = process.env.LIVEKIT_API_KEY;
  const apiSecret = process.env.LIVEKIT_API_SECRET;
  const wsUrl = process.env.LIVEKIT_URL;

  if (!apiKey || !apiSecret || !wsUrl) {
    return NextResponse.json({ error: 'Server misconfigured' }, { status: 500 });
  }

  const at = new AccessToken(apiKey, apiSecret, { identity: username });
  at.addGrant({ room, roomJoin: true, canPublish: true, canSubscribe: true });

  return NextResponse.json(
    { token: await at.toJwt() },
    { headers: { "Cache-Control": "no-store" } },
  );
}
```

Weitere page erforderlich

Make a new file at `/app/room/page.tsx` with the following content:

```
'use client';

import {
  ControlBar,
  GridLayout,
  ParticipantTile,
  RoomAudioRenderer,
  useTracks,
  RoomContext,
} from '@livekit/components-react';
import { Room, Track } from 'livekit-client';
import '@livekit/components-styles';
import { useEffect, useState } from 'react';

export default function Page() {
  // TODO: get user input for room and name
  const room = 'quickstart-room';
  const name = 'quickstart-user';

const [token, setToken] = useState('');

  const [roomInstance] = useState(() => new Room({
    // Optimize video quality for each participant's screen
    adaptiveStream: true,
    // Enable automatic audio/video quality optimization
    dynacast: true,
  }));

  useEffect(() => {
    let mounted = true;
    (async () => {
      try {
        const resp = await fetch(`/api/token?room=${room}&username=${name}
`);
        const data = await resp.json();
        if (!mounted) return;
        if (data.token) {

setToken(data.token);

          await
roomInstance.connect(process.env.NEXT_PUBLIC_LIVEKIT_URL,
data.token);
        }
      } catch (e) {
        console.error(e);
      }
    })();

    return () => {
      mounted = false;
      roomInstance.disconnect();
    };
  }, [roomInstance]);

  if (token === '') {
    return <div>Getting token...</div>;
  }

  return (
    <RoomContext.Provider value={roomInstance}>
      <div data-lk-theme="default" style={{ height: '100dvh' }}>
        {/* Your custom component with basic video conferencing functionality.
*/}
        <MyVideoConference />
        {/* The RoomAudioRenderer takes care of room-wide audio for you. */}
        <RoomAudioRenderer />
        {/* Controls for the user to start/stop audio, video, and screen share
tracks */}
        <ControlBar />
      </div>
    </RoomContext.Provider>
  );
}

function MyVideoConference() {
  // `useTracks` returns all camera and screen share tracks. If a user
  // joins without a published camera track, a placeholder track is returned.
  const tracks = useTracks(
    [
      { source: Track.Source.Camera, withPlaceholder: true },
      { source: Track.Source.ScreenShare, withPlaceholder: false },
    ],
    { onlySubscribed: false },
  );
  return (
    <GridLayout tracks={tracks} style={{ height: 'calc(100vh - var(--lk-control-
bar-height))' }}>
      {/* The GridLayout accepts zero or one child. The child is used
      as a template to render all passed in tracks. */}
      <ParticipantTile />
    </GridLayout>
  );
}
```

Im terminal installieren - ACHTUNG das steht NICHT in den DOCS:

```
% yarn add LiveKit-client
```

WICHTIGER SCHRITT FÜR .env in sunriza26:

GANZ WICHTIG

```
NEXT_PUBLIC_LIVEKIT_URL=LIVEKIT_URL
```

DH in unserem Fall

```
NEXT_PUBLIC_LIVEKIT_URL=wss://sunriza26-g5a1is22.livekit.cloud
```